FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

FEATURE SELECTION METHODS USING LATENT FACTOR MODELS FOR HIGH-DIMENSIONAL DATA

By

RITTWIKA KANSABANIK

A Dissertation submitted to the
Department of Statistics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2025

Rittwika Kansabanik defended this dissertation on November 7, 2025.
The members of the supervisory committee were:

Adrian Barbu
Professor Directing Dissertation

Sonia Haiduc
University Representative

Xin Henry Zhang
Committee Member

Joshua Loyal
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

To Ma, Baba, Dida and Sayan

ACKNOWLEDGMENTS

I am deeply indebted to my Ph.D. advisor, Prof. Adrian Barbu, whose unwavering guidance, insightful feedback, and boundless patience and support were instrumental in shaping this research. His expertise and encouragement pushed me through every challenge, and I am profoundly grateful for his mentorship, which has been invaluable not only in this academic endeavor but also in my personal growth as a researcher. The lessons I have learned from him will undoubtedly influence my entire future career, and for that, I am deeply appreciative.

I sincerely thank my current and former esteemed Ph.D. committee members: Prof. Sonia Haiduc, Prof. Xin Henry Zhang, Prof. Joshua Loyal, Prof. Yiyuan She, and Prof. Chao Huang. Their guidance, suggestions, support, and collaboration throughout this research and during my defense greatly enhanced the quality of this dissertation. I am grateful for the opportunity to learn from their collective wisdom and expertise.

To Ananya, who stood by me during the highs and lows of this doctoral journey, I am incredibly grateful to have you in my life. Your endless support, encouragement, and occasional humor provided the much-needed balance during this intense and demanding time. Your belief in me was a constant source of motivation, and I cherish the moments of laughter and camaraderie that helped me get through the challenges. To my other friends in Tallahassee, Arnab, Rajarshi, Durbadal, Sourita, Tania, and in India—Asmita, Sayan, Debosmita, Rahul, Sen, Abhishek, Soumyajyoti, Shubhomoy, Mainak da, Debojyoti da, Shromona di, Swagata di, Sayan da, Drik da—your friendship has been a continuous source of joy and strength, offering solace during tough times and magnifying the triumphs, even from half a world away. Thank you for always being there and for cheering me on.

I am profoundly thankful to the senior students and now graduates of the department —Sudipto da, Aditi, and Sayantika di —for their kindness, patience, and guidance, which made my transition into the Ph.D. program seamless and welcoming. From sharing insights about coursework to providing tips on navigating the complexities of academic life, their support has been immeasurable.

Words cannot fully express my gratitude for my partner, Sayan, whose love and steadfast support have been my anchor throughout this long and often tumultuous journey. He was the one who listened patiently through my endless explanations of complex statistical concepts, celebrated the smallest breakthroughs, and offered perspective and courage when self-doubt felt overwhelming.

In moments of intense frustration and exhaustion, when I broke down in tears under the weight of challenges, he was the calm presence who took care of me, helping me find the strength to continue. His unwavering belief in my abilities, often stronger than my own, was a constant reminder of why I started this path. This achievement is as much a testament to his patience, sacrifice, and understanding as it is to my own efforts, and I am eternally thankful for having him by my side.

Finally, my deepest and most heartfelt gratitude goes to my family—my parents, Baba and Ma, and my grandmother, Dida. This dissertation, and indeed my entire academic journey, is built upon the foundation of their love, encouragement, and countless sacrifices. To my Baba, for instilling in me a drive for knowledge and a belief that no goal was ever out of reach. To my Ma, for being my greatest cheerleader, my source of boundless emotional strength, and for all the late-night calls that bridged the thousands of miles between us. To my Dida, for her unconditional love and blessings that have been a guiding light throughout my life. Words cannot do justice to the immensely significant role you have all played in every achievement. Your unwavering faith in my abilities, even during times when I doubted myself, has been the very bedrock of my determination and has propelled me forward across every ocean and every challenge. I am grateful to my entire family for this immense faith in my potential, which fueled my resolve to pursue my dreams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\mathbf{\Gamma}$, $\mathbf{M}$	Bold uppercase letters for matrices
$\boldsymbol{\gamma}$, $\mathbf{x}$	Bold lowercase letters for vectors
$\mathbf{x}[i]$	i^{th} component of vector $\mathbf{x}$
$\mathbf{M}[\mathcal{I}, \mathcal{J}]$	A sub-matrix of $\mathbf{M}$ with rows and columns indexed by $\mathcal{J}$ and $\mathcal{I}$
$\mathbf{M}[\mathcal{J}]$	A sub-matrix of $\mathbf{M}$ with all rows and columns indexed by $\mathcal{J}$
$\boldsymbol{M_{ij}}$	The (i, j)-th element of $\mathbf{M}$
$\boldsymbol{M_{i\cdot}}$	The i-th row of matrix $\mathbf{M}$
$\boldsymbol{M_{\cdot j}}$	The j-th column of matrix $\mathbf{M}$
$\mathrm{tr}(\boldsymbol{M})$	The trace of a matrix $\mathbf{M}$
$\det(\mathbf{M})$	The determinant of a matrix $\mathbf{M}$
$\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$	A diagonal matrix with diagonal entries: $\lambda_1, \lambda_2, \cdots, \lambda_d$
$D(\boldsymbol{M})$	Diagonal matrix that has the same diagonal elements as $\mathbf{M}$
$\Delta(\boldsymbol{M})$	$\Delta(\mathbf{M}) = \mathbf{M} - D(\mathbf{M})$
$\hat{\boldsymbol{M}}$	The estimated matrix $\boldsymbol{M}$
$\mathbf{I}_q$	The identity matrix of order $q \times q$
$\|\|\boldsymbol{M}\|\|_F$	Frobenius norm of the matrix $\boldsymbol{M}$
$\|\|\boldsymbol{M}\|\|_{2,0}$	The number of non-zero rows in $\boldsymbol{M}$
$\|\|\boldsymbol{M}\|\|_0$	The number of non-zero emtries in $\boldsymbol{M}$
$\mathbf{M}^+$	The Moore–Penrose inverse of $\boldsymbol{M}$
$\|\|\mathbf{x}\|\|_2$	L_2 norm of a vector $\mathbf{x}$
$\|\|\mathbf{x}\|\|_\infty$	L_∞ norm of a vector $\mathbf{x}$
$< \mathbf{x}_1, \mathbf{x}_2 >$	The inner product between two vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$
d	The dimension of the data
$\mathcal{F}_k$	Set of features with cardinality k
f_k	Feature k
i, n, p	Scalars denoted by lower case letters

ABSTRACT

This research provides a thorough analysis of feature selection methods in machine learning. The study addresses challenges associated with high-dimensional data and aims to alleviate the curse of dimensionality. The research is conducted on enhancing model performance through feature selection techniques. It systematically reviews existing feature selection approaches, including both supervised and unsupervised methods. New strategies are proposed to improve robustness and create sparsity in the feature selection process. Additionally, the research emphasizes the critical evaluation of these methods within a multi-class classification framework, utilizing both simulated and real-world datasets. Key contributions of the study include the development of a signal-to-noise ratio (SNR)-based feature selection technique, the theoretical investigation of feature recovery guarantees, the proposal of robust outlier handling methods, the integration of per-class feature selection for multi-class classification, and the execution of comprehensive experiments to confirm the effectiveness and robustness of the proposed methods.

CHAPTER 1

INTRODUCTION

Feature selection involves identifying a relevant subset of features from a larger set to address the challenge of dealing with too many dimensions in data. Features are individual, measurable properties of what is being studied. Machine learning algorithms utilize these features for classification, regression, and other purposes. As machine learning has advanced, the number of features used has also grown.

Machine learning methods are expected to perform better when they have more information. However, dealing with high-dimensional data poses challenges known as the curse of dimensionality. As the number of features increases, issues such as training time, algorithmic complexity, storage space, and noise in datasets can worsen. Noise can be referred to as the set of variables that do not influence the target variable and may introduce bias in the prediction, or as the set of dependent variables that provide no additional information [24, 97]. The performance of a classifier depends on the interrelationship between the number of samples and the number of features used. Interestingly, adding more features to the dataset can improve accuracy while the signal dominates the noise. Beyond that point, the model accuracy reduces. This phenomenon is called peaking [56].

Therefore, dimensionality reduction techniques have become popular. They involve reducing the number of features in a model, potentially through transformation methods. Feature selection is a type of dimensionality reduction that removes features from the model's inputs. The key distinction is that dimensionality reduction may require all data sources to transform and reduce features, whereas feature selection avoids irrelevant data collection while still providing good predictive results. It has several advantages, as suggested by [13], such as:

- Increase the speed and scalability of the model, which are desirable traits for large-scale computation.
- Removes noise and nuisance dimensions from the data to recover genuine signals with high probability.

Several challenges arise in constructing feature selection algorithms for big data, as noted in [108] and [13]. Some of them are the following:

- The aim should be to select a smaller set of features while maintaining model accuracy, ensuring that the accuracy of the chosen subset does not drop significantly compared to the model trained on all features.
- The class distribution obtained for the selected features should closely resemble the original class distribution, considering all features.
- Algorithms with simple implementations are preferred to avoid overfitting and ad-hoc designs.
- Consideration of nonlinear patterns of the features in the algorithm is also desirable.

Feature selection is integral to both supervised and unsupervised learning paradigms. In supervised learning, the primary aim is to differentiate between data points belonging to distinct classes (classification) or to provide accurate predictions of regression targets. Conversely, unsupervised feature selection addresses challenges in clustering. Instead of relying on labels, it seeks alternative metrics to measure the significance of available features. Feature selection can be achieved through various approaches: they may operate independently of learning algorithms (filter methods), depend on learning algorithms iteratively to improve the quality of selected features (wrapper methods), or integrate the feature selection phase into supervised/unsupervised learning algorithms (embedded methods). Ultimately, in supervised setups, the trained classifier or regression model employs the selected features to predict class labels or regression targets for test data points. In contrast, in unsupervised scenarios, it provides the cluster structure of all data samples based on the selected features using a standard clustering algorithm [40, 74].

The usual approach is to optimize a margin-maximizing loss function, which scales as $O(C)$ with the number of classes C. Our work emphasizes individual class modeling, independently leveraging a generative model and feature selection for each class. This class-specific modeling sets our method apart from existing feature selection techniques for the following reasons:

- It captures the unique characteristics and distribution by tailoring the model to each class.
- The model for each class is wrapped tightly around the observations of that class, which allows the introduction of new classes without retraining the existing class models and scales as $O(1)$.
- Additionally, preserving learned parameters for each class mitigates the risk of catastrophic forgetting when new data is introduced.

In our first step, we propose using the signal-to-noise ratio (SNR) as a feature selection criterion, where the signal represents relevant information that contributes to accurate predictions, and the

noise represents irrelevant data. SNR quantifies the strength of the signal relative to noise, with higher SNR features being more effective at distinguishing classes. Eliminating low SNR features enhances computational efficiency and model interpretability. We employ low rank generative models for individual class modeling and SNR estimation.

Recently, low-rank models have attracted considerable attention for feature selection, owing to their capacity to capture underlying latent structures and disentangle informative signals from noise. For instance, low-rank learning methods have been proposed for multi-label feature selection in [72], demonstrating that low-dimensional latent representations can enhance discriminative performance. Previously, factor analysis—a classical generative model—was applied to feature selection in the context of Alzheimer's disease diagnosis [96], where factor loadings identified relevant brain regions. Collectively, these studies underscore the potential of low-rank and latent factor models for robust feature extraction. However, they often lack formal non-asymptotic theoretical guarantees. This thesis also takes an initial step toward analyzing the asymptotic properties and finite-sample bounds for the signal variance, noise variance, and SNR estimation error, which are crucial for guaranteeing true feature recovery. True feature recovery guarantees ensure that as we receive infinitely many observations, the estimated SNRs should converge to the true counterparts.

In the next step, we also experiment with reduced-rank regression methods with sparsity constraints to perform feature selection. We also propose a novel robust loss-based rank optimization further to reduce the impact of detrimental outliers in the dataset. In the simulation setup, we have demonstrated how robust loss-based methods can effectively handle outliers when traditional l_2-norm-based approaches completely break down. We have also employed these methods for class-specific feature selection and conducted a performance comparison on real datasets with the previously mentioned low-rank generative models.

In our next step, we apply the selected set of features within a multi-class classification framework. Within this framework, the feature selection process is carried out independently for each class. Subsequently, the selected features are used to compute the Bayesian probability for each existing class for a new observation. The classification of the latest observation is then determined by assigning it to the class with the highest posterior probability. This method is immune to catastrophic forgetting. Therefore, we have also employed our class-specific feature selection method for class incremental learning using real data sets and have performed a comparative analysis with contemporary techniques.

In our experiments, we analyzed the feature selection abilities and classification accuracy of these techniques using a simulated dataset. Additionally, we have evaluated their classification performance across different computer vision datasets, including CIFAR-10 [65] and CIFAR-100 [66], each with 60,000 training images and 10 and 100 categories, respectively, and ImageNet-1k [94] with 1.2 million training images and 1000 classes.

This dissertation presents a comprehensive investigation into feature selection, culminating in a novel framework that is robust, scalable, and theoretically sound. The research systematically reviews existing paradigms while introducing new strategies to advance the state of the art. The key contributions of this work are summarized as follows:

- **A Novel SNR-Based Feature Selection Framework:** It introduces a feature selection method based on the Signal-to-Noise Ratio (SNR) criterion for a class of low-rank generative models, including Probabilistic PCA (PPCA) [110], Latent Factor Analysis (LFA) [35], ELF [59], and Heteroskedastic PCA [122].
- **Rigorous Theoretical Guarantees:** It provides a detailed asymptotic and non-asymptotic analysis for the proposed feature selection methods. This work establishes theoretical guarantees for true feature recovery under certain assumptions, providing a principled foundation that moves beyond heuristic approaches.
- **A Robust Method for Handling Outliers:** Recognizing that real-world datasets are often contaminated with outliers, a robust feature selection method is developed. This approach incorporates a sparsity constraint and robust loss functions to effectively select influential features even in the presence of data that could otherwise degrade model performance.
- **A Scalable Framework for Multi-Class and Incremental Learning:** It shows how to apply the proposed feature selection method to multi-class classification, resulting in a class-incremental learning method that is structurally immune to catastrophic forgetting. This allows for the seamless addition of new classes without retraining on the entire dataset.
- **Comprehensive Experimental Validation:** The efficacy and robustness of the proposed methods are rigorously validated through comprehensive experiments on both simulated data and large-scale, real-world computer vision benchmarks, including CIFAR-10 [65], CIFAR-100 [66], and ImageNet-1k [94].
- **Comparative Performance Analysis:** The proposed method is compared against standard linear model-based and recent state-of-the-art feature selection methods. The results demonstrate that our approach significantly outperforms classic methods by a wide margin and shows competitive performance in class-incremental learning setups, validating its practical applicability.

CHAPTER 2

SIGNAL TO NOISE RATIO (SNR) FOR FEATURE SELECTION

This chapter introduces a feature selection technique that uses SNR as the criterion. This method can be applied to various low-rank generative models, such as Probabilistic PCA and Latent Factor Analysis. First, we describe these methods and their parameter estimation processes. We then use these estimates to calculate the SNRs.

2.1 Related Work

2.1.1 PCA & LFA-Based and Hybrid Feature Selection

While many techniques leverage Principal Component Analysis (PCA) for dimensionality reduction, these approaches to feature selection differ significantly from the proposed SNR-based method. Boutsidis et al. [18] focused on selecting a representative subset of features that preserves the variance captured by top eigenfeatures. Our SNR-based method, however, evaluates each feature individually for its discriminatory power rather than selecting a collective subset to represent the whole. The work by Niu and Qiu [82] on weighted PCA is conceptually extended by the SNR approach, which formalizes the weighting by using the inverse of the noise covariance to distinguish meaningful signals from noise systematically.

Several hybrid methods use PCA as a preliminary step before applying other selection techniques. For instance, [112] employed a two-stage method combining Information Gain and a Genetic Algorithm with PCA. Similarly, Ahmad [4], Alomari et al. [5], and Pushpalatha et al. [85] used evolutionary algorithms such as the GA, Grey Wolf Optimizer, and ReliefF to refine a PCA-reduced feature set. These multi-step approaches separate dimensionality reduction from feature selection, whereas our SNR-based method provides an integrated solution in which the selection criterion is inherent to the model.

Supervised PCA variants also differ in their core mechanism. Sharifzadeh et al. [99] and Rahmat et al. [88] incorporate supervision by identifying features that are highly dependent on a response variable. In contrast, our SNR-based method models each class independently and selects features

based on their ability to represent the unique characteristics of that class, not just their correlation with an output variable.

2.1.2 Methods Based on Latent Factor Models

Feature selection using latent factor models has also seen diverse approaches. The Sparse Estimation of Latent Factors (SELF) framework proposed by Aziz [8] achieves feature selection by imposing sparsity directly on the model's transformation matrix $\mathbf{W}$. The SNR-based method diverges from this by removing such structural constraints and instead using the signal-to-noise ratio as a post-hoc criterion to rank features based on the learned model.

The work of Abbas and Sivaswamy [1] utilized latent factors to extract influential low-dimensional features from medical images, followed by classification using Mahalanobis distance. This constitutes a feature-extraction approach, creating new features rather than a feature-selection method that ranks and chooses from the original set of features, which is the focus of the SNR method. Similarly, Townes et al. [111] proposed a latent factor model for single-cell RNA-sequencing data that ranks genes based on deviance, a metric tailored to count data. Our SNR-based method is more general, defining the signal and noise based on the variance explained by the generative model, making it applicable across various data types.

In conclusion, while PCA- and LFA-based methods often rely on hybrid frameworks or structural constraints for feature selection, the SNR-based approach offers a distinct, unified methodology. It evaluates features on a class-by-class basis using an intrinsic, theoretically grounded measure of their signal content.

2.1.3 Signal-to-Noise Ratio (SNR)–Based Feature Selection

The Signal-to-Noise Ratio (SNR) is one of the simplest and most interpretable measures for identifying discriminative features. It quantifies how strongly a feature separates classes relative to within-class variation, making it an effective filter criterion in high-dimensional data.

The use of SNR as a feature selection criterion dates back to early work in neural networks. [15] introduced an SNR-based saliency measure to identify and prune noisy inputs during training by comparing each input's contribution to that of a random noise feature. This approach proved effective for dimensionality reduction on benchmark datasets, outperforming standard PCA-based methods. Shortly after, the now-classic SNR formulation, $SNR = (\mu_1 - \mu_2)/(\sigma_1 + \sigma_2)$, was employed

by [48] as a screening criterion for selecting features in binary classification tasks with probabilistic neural networks, where (μ_i, σ_i) represent the mean and standard deviation of a feature for class i.

The utility of SNR became particularly evident in bioinformatics, where microarray and gene expression data are characterized by high dimensionality and significant noise. To improve feature ranking, [79] and later [95] proposed hybrid clustering frameworks where SNR was used to select the most informative genes within each cluster, effectively reducing redundancy and enhancing classification accuracy.

In recent years, SNR-based methods have been hybridized with other machine learning techniques to enhance their robustness and applicability across diverse domains. In engineering, [41] used a PCA-based signal subspace approach to improve the SNR of noisy vibration signals for early fault detection in ball bearings. For wireless positioning systems, [83] developed SNR-driven feature reduction techniques to minimize model complexity while maintaining predictive power in low-SNR environments.

In genomics, Weighted SNR (WSNR) methods have been developed, such as the one by [42], which integrates SNR scores with Support Vector Machine (SVM) weights to emphasize the most discriminative genes. To mitigate the influence of outliers in skewed datasets, [53] combined SNR scores with the Mood median test, creating a robust "Md-score" that balances class separation and statistical significance. Beyond classification, SNR has been adapted for nonlinear regression in physical systems. [16] introduced an ANN-SNR method with confidence interval stopping rules to predict concrete shear strength, achieving high accuracy with a significantly reduced feature set.

The existing body of work demonstrates that SNR is a flexible and interpretable feature selection strategy. However, these methods are often heuristic, lack theoretical guarantees, and typically use all available data to compute a single SNR value for each feature. Our proposed approach introduces several key novelties that address these limitations.

First, it defines a different SNR criterion based on the parameters of a generative latent factor model, where the signal is captured by the variance explained by the latent factors, and the noise is the unexplained variance. Second, it proposes a class-based feature selection paradigm in which the SNR for each feature is computed using only the data available for that class. This class-specific modeling makes the approach highly scalable, naturally suited for class-incremental learning, and capable of capturing the unique characteristics of each class.

Finally —and most significantly —our work takes a firm step toward establishing a rigorous theoretical foundation. We analyze the asymptotic properties of the parameter estimates used

to compute the SNRs and provide a path to deriving non-asymptotic probability bounds for the estimated signal, noise, and SNR values. This provides a theoretical basis for SNR-based feature selection in latent factor models, moving the field beyond heuristic applications toward a more principled, scalable, and theoretically grounded methodology.

2.2 Low-rank Generative Models

In this section, we are going to describe the four different methods based on low-rank generative models that are included in this study, namely Probabilistic PCA (PPCA) [110], Latent Factor Analysis (LFA)[35], Heteroskedastic PCA (HeteroPCA) [122] and Estimation of Latent Factors (ELF). We have introduced the last method in [59], which is a nonparametric version of LFA.

PPCA, LFA, and our newly introduced method, ELF, share the same model structure but have different assumptions associated with their model parameters. The model aims to find a relationship between the observed $\mathbf{x} \in \mathbb{R}^d$ and a hidden set of variables (latent variables) $\boldsymbol{\gamma} \in \mathbb{R}^r$ with $r << d$ and assumes the latent factors and noise variables are independent of each other. It is as follows:

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{W}\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \quad \text{with } E(\boldsymbol{\epsilon}) = \mathbf{0} \text{ and } \text{var}(\boldsymbol{\epsilon}) = \boldsymbol{\Psi}. \tag{2.1}$$

The PPCA and LFA methods assume that $\boldsymbol{\gamma} \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ and that the noise variable $\boldsymbol{\epsilon} \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$. It can be easily verified that for these two methods:

$$\mathbf{x}|\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{W}\boldsymbol{\gamma} + \boldsymbol{\mu}, \boldsymbol{\Psi}), \text{ and by integration,} \tag{2.2}$$

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi}. \tag{2.3}$$

Conversely, ELF does not make distributional assumptions about the parameters it estimates. ELF assumes $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \cdots, \boldsymbol{\gamma}_n)^T$ to be semi-orthogonal ($\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \mathbf{I}_r$). LFA and ELF, while sharing similar goals with PPCA, assume distinct noise variances across dimensions.

$$\boldsymbol{\Psi} = \begin{cases} \sigma^2\mathbf{I}_d & \text{for PPCA,} \\ \text{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2) & \text{otherwise.} \end{cases} \tag{2.4}$$

$\boldsymbol{\mu}$ has been treated as a constant vector in the model (2.1) and estimated as: $\boldsymbol{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$.

2.2.1 Parameter estimation for PPCA and LFA

Due to the isotropic nature of the $\mathbf{\Psi}$ in PPCA, a closed form of the Maximum Likelihood (ML) estimates of the PPCA model parameters $(\mathbf{W}, \sigma^2)$ has been derived in [110]. It is as follows:

$$\sigma^2_{ML} = \frac{1}{d-r} \sum_{j=r+1}^{d} l_j, \tag{2.5}$$

$$\mathbf{W}_{ML} = \mathbf{U}_r(\mathbf{S}_r - \sigma^2_{ML}\mathbf{I}_r)^{0.5}\mathbf{R}, \tag{2.6}$$

where l_j is the j^{th} largest eigenvalue and $\mathbf{U}_r$ consists of the first r principal eigenvectors of the sample covariance matrix, $\hat{\mathbf{\Sigma}}$, the matrix $\mathbf{S}_r = \text{diag}(l_1, l_2, \cdots, l_r)$, while $\mathbf{R}$ is an arbitrary $r \times r$ orthogonal rotation matrix.

As discussed in [110], an essential capability of PPCA is density modeling, whether through individual or mixture models. PPCA can manage the model's complexity by selecting a rank r with $r \ll d$. This choice helps limit the number of parameters used to define the covariance in the high-dimensional space. In situations where employing fully parameterized covariance matrices would lead to excessive under-constraint due to data dimensionality, this approach becomes useful. It allows one to avoid problems that can arise from constraining the covariance to be diagonal or spherical, which may be inappropriate for specific datasets. Furthermore, when it comes to classification tasks, modeling the densities associated with different classes makes sense even when the data dimensionality is quite large.

Latent Factor Analysis (LFA) is a multivariate statistical technique commonly used for dimensionality reduction. This analytical approach shares significant kinship with PCA, which aims to find orthogonal components (principal components) that maximize the variance in the data. In contrast, LFA seeks to discover factors that account for observed variations but does not necessarily require orthogonality.

LFA parameters $(\mathbf{W}, \mathbf{\Psi})$ can be estimated using an EM algorithm due to [35].

Theorem 1 (due to [35]). *Assume that the data has been properly centralized and let* $\boldsymbol{\beta} = \hat{\mathbf{W}}^T(\hat{\mathbf{\Psi}} + \hat{\mathbf{W}}\hat{\mathbf{W}}^T)^{-1}$. *The EM updates of* $(\hat{\mathbf{W}}, \hat{\mathbf{\Psi}})$ *for LFA are:*

- ***E-step****: Compute* $E(\boldsymbol{\gamma}|\mathbf{x}_i)$ *and* $E(\boldsymbol{\gamma}\boldsymbol{\gamma}^T|\mathbf{x}_i)$ *for each data point* $\mathbf{x}_i$ *as follows:*

$$E(\boldsymbol{\gamma}|\mathbf{x}_i) = \boldsymbol{\beta}\mathbf{x}_i,\ E(\boldsymbol{\gamma}\boldsymbol{\gamma}^T|\mathbf{x}_i) = \mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}^T.$$

- ***M-step*** *Update the LFA parameters as:*

$$\hat{\mathbf{W}}_{new} = \sum_{i=1}^{n} \mathbf{x}_i E(\boldsymbol{\gamma}|\mathbf{x}_i)^T (\sum_{i=1}^{n} E(\boldsymbol{\gamma}\boldsymbol{\gamma}^T|\mathbf{x}_i))^{-1}, \tag{2.7}$$

$$\boldsymbol{\Psi}_{new} = \frac{1}{n} D(\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T - \hat{\mathbf{W}}_{new} E(\boldsymbol{\gamma}|\mathbf{x}_i)\mathbf{x}_i^T). \tag{2.8}$$

2.2.2 Parameter estimation for ELF.

ELF estimates the model parameters $(\boldsymbol{\Gamma}, \mathbf{W})$ by optimizing the following:

$$(\hat{\mathbf{W}}_{ELF}, \hat{\boldsymbol{\Gamma}}_{ELF}) = \underset{(\boldsymbol{\Gamma},\mathbf{W}),\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}=\mathbf{I}_r}{\operatorname{argmin}} \|(\mathbf{X} - \boldsymbol{\Gamma}\mathbf{W}^T)\boldsymbol{\Psi}^{-\frac{1}{2}}\|_F^2, \tag{2.9}$$

where $\mathbf{X} = (\mathbf{x}_1 - \boldsymbol{\mu}_{ML}, \mathbf{x}_2 - \boldsymbol{\mu}_{ML}, \cdots, \mathbf{x}_n - \boldsymbol{\mu}_{ML})^T$, i.e. properly centralized. We use $\boldsymbol{\Psi}^{-\frac{1}{2}}$ as feature weights in (2.9) to reduce the impact of features with significant unexplained noise variance, thereby significantly improving model accuracy. During model training, we estimate: $\hat{\sigma}_j^2 = \|\hat{\mathbf{X}}_{\cdot j} - \mathbf{X}_{\cdot j}\|_2^2/(n-1)$ and employ $(\hat{\sigma}_j^2)^{-\frac{1}{2}}$ as j^{th} feature weight for the estimation process.

To perform the minimization in (2.9), in every iteration, we first estimate $(\hat{\mathbf{W}}, \hat{\boldsymbol{\Gamma}})$ without the constraint on $\boldsymbol{\Gamma}$ using Theorem 2 and then adjust the estimated parameters to satisfy the constraint using the Proposition 1 below.

Theorem 2. *The ELF objective* (2.9) *without the constraint* $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \mathbf{I}_r$ *is minimized w.r.t* $\boldsymbol{\Gamma}$ *and* $\mathbf{W}$ *by*

$$\hat{\boldsymbol{\Gamma}} = \mathbf{X}\boldsymbol{\Psi}^{-1}\mathbf{W}(\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1} \textit{and } \hat{\mathbf{W}} = \mathbf{X}^T\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}. \tag{2.10}$$

Proof. Let $l(\boldsymbol{\Gamma}) = \|(\mathbf{X} - \boldsymbol{\Gamma}\mathbf{W}^T)\sqrt{\boldsymbol{\Psi}^{-1}}\|_F^2$. Then

$$\begin{aligned}
\arg\min_{\boldsymbol{\Gamma}} l(\boldsymbol{\Gamma}) &= \arg\min_{\boldsymbol{\Gamma}} \|\mathbf{X}\sqrt{\boldsymbol{\Psi}^{-1}} - \boldsymbol{\Gamma}\mathbf{W}^T\sqrt{\boldsymbol{\Psi}^{-1}}\|_F^2 \\
&= \arg\min_{\boldsymbol{\Gamma}} \operatorname{Tr}((\mathbf{X}\sqrt{\boldsymbol{\Psi}^{-1}} - \boldsymbol{\Gamma}\mathbf{W}^T\sqrt{\boldsymbol{\Psi}^{-1}})^T(\mathbf{X}\sqrt{\boldsymbol{\Psi}^{-1}} - \boldsymbol{\Gamma}\mathbf{W}^T\sqrt{\boldsymbol{\Psi}^{-1}})) \\
&= \arg\min_{\boldsymbol{\Gamma}} \operatorname{Tr}(\sqrt{\boldsymbol{\Psi}^{-1}}\mathbf{X}^T\mathbf{X}\sqrt{\boldsymbol{\Psi}^{-1}} - 2\sqrt{\boldsymbol{\Psi}^{-1}}\mathbf{X}^T\boldsymbol{\Gamma}\mathbf{W}^T\sqrt{\boldsymbol{\Psi}^{-1}} + \sqrt{\boldsymbol{\Psi}^{-1}}\mathbf{W}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{W}^T\sqrt{\boldsymbol{\Psi}^{-1}}) \\
&= \arg\min_{\boldsymbol{\Gamma}} \operatorname{Tr}(\boldsymbol{\Psi}^{-1}\mathbf{X}^T\mathbf{X} - 2\boldsymbol{\Psi}^{-1}\mathbf{X}^T\boldsymbol{\Gamma}\mathbf{W}^T + \boldsymbol{\Psi}^{-1}\mathbf{W}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{W}^T) \\
&= \arg\min_{\boldsymbol{\Gamma}} \operatorname{Tr}(-2\boldsymbol{\Psi}^{-1}\mathbf{X}^T\boldsymbol{\Gamma}\mathbf{W}^T + \boldsymbol{\Psi}^{-1}\mathbf{W}\boldsymbol{\Gamma}'\boldsymbol{\Gamma}\mathbf{W}^T) \\
&= \arg\min_{\boldsymbol{\Gamma}} \operatorname{Tr}(-2\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{X}^T\boldsymbol{\Gamma} + \boldsymbol{\Gamma}^T\boldsymbol{\Psi}^{-1}\mathbf{W}\boldsymbol{\Gamma}^T).
\end{aligned}$$

$$\frac{\partial l(\boldsymbol{\Gamma})}{\partial \boldsymbol{\Gamma}} = \frac{\partial}{\partial \boldsymbol{\Gamma}} \operatorname{Tr}(-2\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{X}^T\boldsymbol{\Gamma} + \boldsymbol{\Gamma}^T\boldsymbol{\Psi}^{-1}\mathbf{W}\boldsymbol{\Gamma}^T) = -2\mathbf{X}\boldsymbol{\Psi}^{-1}\mathbf{W} + 2\boldsymbol{\Gamma}\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W}.$$

$$\frac{\partial l(\boldsymbol{\Gamma})}{\partial \boldsymbol{\Gamma}} = 0 \implies \boldsymbol{\Gamma} = \mathbf{X}\boldsymbol{\Psi}^{-1}\mathbf{W}(\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1}.$$

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{j=1}^{d} \frac{\|\mathbf{X}_{\cdot j} - \boldsymbol{\Gamma}\mathbf{W}_{j\cdot}^T\|^2}{\sigma_j^2},$$

which is minimized individually for each $\mathbf{W}_j$ as $\mathbf{W}_{j\cdot}^T = (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\mathbf{X}_{\cdot j}$, which gives the result. □

Proposition 1. *If* $\mathbf{UDV}^T = \boldsymbol{\Gamma}$ *is the SVD of* $\boldsymbol{\Gamma}$*, then* $\boldsymbol{\Gamma}_1 = \mathbf{U}$*, and* $\mathbf{W}_1 = \mathbf{WVD}$ *satisfy* $\boldsymbol{\Gamma}_1\mathbf{W}_1^T = \boldsymbol{\Gamma}\mathbf{W}^T$ *along with* $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = \mathbf{I}_r$*.*

Proof. It is easy to verify that $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = \mathbf{I}_r$.□

The $\hat{\mathbf{W}}$ produced in (2.10) does not depend on the feature weights $\boldsymbol{\Psi}$. Algorithm 1 summarizes the iterative estimation procedure.

Algorithm 1: Parameter Estimation for ELF

Input: $\mathbf{X}_{n\times d}$, T (number of iterations) and m

Output: $\hat{\mathbf{W}}$ and $\hat{\boldsymbol{\Psi}}$

Initialize

- feature weight matrix $\boldsymbol{\Psi} = \boldsymbol{I_d}$
- $\boldsymbol{\Gamma}$ as the first r principal components of $\mathbf{X}$
- $\mathbf{W}$ as the first r loading vectors from PCA of $\mathbf{X}$

for $t = 1$ *to* T **do**

Update $\mathbf{W}$ as $\mathbf{W} = \mathbf{X}^T\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}$

Update $\boldsymbol{\Gamma}$ as $\boldsymbol{\Gamma} = \mathbf{X}\boldsymbol{\Psi}^{-1}\mathbf{W}(\mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1}$

Perform SVD on $\boldsymbol{\Gamma}$, $\mathbf{U}_{\boldsymbol{\Gamma}}\mathbf{D}_{\boldsymbol{\Gamma}}\mathbf{V}_{\boldsymbol{\Gamma}}^T = \boldsymbol{\Gamma}$

Update $\boldsymbol{\Gamma} = \mathbf{U}_{\boldsymbol{\Gamma}}$ and $\mathbf{W} = \mathbf{W}\mathbf{V}_{\boldsymbol{\Gamma}}\mathbf{D}_{\boldsymbol{\Gamma}}$

Update $\boldsymbol{\Psi} = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \cdots \sigma_d^2)$ with $\sigma_i^2 = \operatorname{var}(\mathbf{X}_{\cdot i} - \boldsymbol{\Gamma}\mathbf{W}_{i\cdot}^T)$

Check for convergence: $\|\mathbf{X} - \boldsymbol{\Gamma}\mathbf{W}^T\|_F$ is sufficiently small.

2.2.3 Parameter Estimation for HeteroPCA.

Heteroskedastic PCA [122], also known as Hetero PCA, addresses the issue of performing PCA when the data has heteroskedastic noise, meaning the noise variance differs across dimensions in a spiked covariance model setup. It assumes the following setup:

$$\mathbf{X}_{d\times n} = \mathbf{X}_0 + \boldsymbol{\epsilon}, \qquad E(\mathbf{X}_0) = \boldsymbol{\mu}, \qquad Cov(\mathbf{X}_0) = \boldsymbol{\Sigma}_0, \tag{2.11}$$

$$E(\boldsymbol{\epsilon}) = 0, \qquad \boldsymbol{\Psi} = Cov(\boldsymbol{\epsilon}) = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2). \tag{2.12}$$

Here, $\mathbf{X}_0$ is the noise-free version (signal) of the given data matrix, $\mathbf{X}$. Also, $\boldsymbol{\epsilon}$ and $\mathbf{X}_0$ are independent. $\boldsymbol{\Sigma}_0$ admits rank-r eigen-decomposition $\boldsymbol{\Sigma}_0 = \mathbf{U}\mathbf{D}\mathbf{U}^T$ with $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{D} \in \mathbb{R}^{r\times r}$. The goal is to estimate $\mathbf{U}$.

Though the model is similar to LFA, there is a difference between the objectives of the two models. LFA focuses on finding latent factors that explain the behavior of the observed variables, but Hetero PCA aims to capture the principal components (PCs) (estimate $\mathbf{U}$) of the underlying data($\mathbf{X}_0$), accounting for heteroskedasticity. This is useful when noise levels vary significantly across samples and could bias traditional PCA.

The estimation of $\mathbf{U}$ using the classical PCA is equivalent to the estimation of eigenvectors of the sample covariance matrix $Cov(\mathbf{X}) = \hat{\boldsymbol{\Sigma}}$. Since $E(\hat{\boldsymbol{\Sigma}}) = \boldsymbol{\Sigma}_0 + \boldsymbol{\Psi}$ and $(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2)$ in the diagonals of $\boldsymbol{\Psi}$ are not necessarily same, there will be a significant difference between the principal components of $E(\hat{\boldsymbol{\Sigma}})$ and those of $\boldsymbol{\Sigma}_0$. To cope with the bias on the diagonal elements of the covariance matrix, HeteroPCA iteratively updates the diagonal entries based on the off-diagonals, so that the bias incurred on the diagonal is significantly reduced and more accurate estimation can be achieved. The idea is originally inspired by diagonal deletion SVD [32], which states to set the diagonal of the sample covariance matrix to zero before performing singular value decomposition.

In Algorithm 2, the estimate of $\mathbf{U}$ is iteratively updated by imputing the diagonal entries of the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ by the diagonal entries of its low rank r approximation $\tilde{\mathbf{N}}$, to minimize the following: $\tilde{\mathbf{N}} = \text{argmin}_{\mathbf{N}, r(\mathbf{N})\leq r} \|\Delta(\hat{\boldsymbol{\Sigma}} - \mathbf{N})\|_F^2$.

2.3 Estimation of SNR

The estimated signal-to-noise ratio (SNR) for the available features is computed from the $(\hat{\mathbf{W}}, \hat{\boldsymbol{\Psi}})$ estimates obtained by the different methods. The SNR for the i-th feature is defined as:

$$\boldsymbol{SNR}_i = \frac{\sum_{j=1}^{r} \mathbf{W}_{ij}^2}{\sigma_i^2}, i \in \{1, 2, \cdots, d\}. \tag{2.13}$$

The SNR can be directly calculated using Eq. (2.13) for PPCA, LFA, and ELF methods, due to the assumption of $\text{var}(\gamma) = \mathbf{I}$ or $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \mathbf{I}$. For HeteroPCA, we obtain the r principal loading vectors $\hat{\mathbf{U}}$, corresponding to $\mathbf{X}_0$. To evaluate (2.13) for HeteroPCA, we execute the following steps to obtain $(\hat{\mathbf{W}}, \hat{\sigma}_1^2, \cdots, \hat{\sigma}_d^2)$:

- **Estimation of signal strength:** Our initial estimates are: $\tilde{\boldsymbol{\Gamma}} = \mathbf{X}\hat{\mathbf{U}}$ and $\tilde{\mathbf{W}} = \hat{\mathbf{U}}$. Next, we employ Proposition 1 to obtain semi-orthogonal $\hat{\boldsymbol{\Gamma}}$ and the corresponding $\hat{\mathbf{W}}$. Therefore, $\hat{\mathbf{X}}_0 = \hat{\boldsymbol{\Gamma}}\hat{\mathbf{W}}^T$.

Algorithm 2: Heteroskedastic PCA

Input : $\hat{\mathbf{\Sigma}}$: $Cov(\mathbf{X})$, r : the rank of $\hat{\mathbf{\Sigma}}$, T : maximum number iterations

Output : Estimated rotation matrix $\hat{\mathbf{U}}$, $\hat{\mathbf{\Sigma}}_0$: estimated rank-r approximation of $\mathbf{\Sigma}_0$

Initialize: Initialize by setting the diagonal elements of $\hat{\mathbf{\Sigma}}$ to 0: $\mathbf{N}_{(0)} = \Delta(\hat{\mathbf{\Sigma}})$

for $t = 0$ *to* T **do**

Perform SVD on $\mathbf{N}_{(t)}$ and let $\tilde{\mathbf{N}}_{(t)}$ be the best rank-r approximation:

$$\mathbf{N}_{(t)} = \mathbf{U}_{(t)}\mathbf{D}_{(t)}(\mathbf{V}_{(t)})^T = \sum_i (\lambda_i)_{(t)}(\mathbf{U}_{\cdot i})_{(t)}(\mathbf{V}_{\cdot i})_{(t)}^T, (\lambda_1)_{(t)} \leq (\lambda_2)_{(t)} \leq \cdots (\lambda_d)_{(t)}$$

$$\tilde{\mathbf{N}}_{(t)} = \sum_{i=1}^{r} (\lambda_i)_{(t)}(\mathbf{U}_{\cdot i})_{(t)}(\mathbf{V}_{\cdot i})_{(t)}^T$$

Update $\mathbf{N}_{(t+1)} = D(\tilde{\mathbf{N}}_{(t)}) + \Delta(\mathbf{N}_{(t)})$

Until convergence or maximum number of iterations reached.

The outputs are the first r columns of $\hat{\mathbf{U}}_{(t)}$: $\hat{\mathbf{U}}_{(t)}[\mathcal{J}], \mathcal{J} = \{1, 2, \cdots, r\}$

- **Estimation of noise variance:** Next, the estimation process of $(\sigma_i^2, i = 1, 2, \cdots, d)$ is as follows:

$$\hat{\sigma_i^2} = \|(\hat{\mathbf{X}}_0)_{i\cdot} - \mathbf{X}_{i\cdot}\|_2^2/(n-1). \tag{2.14}$$

The intuition for employing SNRs to identify key features in the latent factor model is based on the assumption that the data originates from a lower-dimensional latent space. The signal is represented as $\mathbf{W}\boldsymbol{\gamma}$ with the assumption $\mathbf{\Gamma}^T\mathbf{\Gamma} = \mathbf{I}_r$. The variance of the corresponding signals is captured by the diagonal elements of $\mathbf{W}\mathbf{W}^T$ or the row sum of squares of $\mathbf{W}$. At the same time, the unexplained noise variance is reflected in the diagonal elements of $\mathbf{\Psi}$. Therefore, features with relatively high SNR values are identified as strongly associated with the latent variables, making them prime candidates for representing objects within specific categories. Once we estimate the SNRs, we perform feature selection by employing a simple thresholding technique, as described in Algorithm 3.

Algorithm 3: SNR based Feature Selection

Input: $(\hat{\mathbf{W}}, \hat{\sigma}_1^2, \cdots, \hat{\sigma}_d^2)$, m

Output: $\mathcal{I}_m$, the indices of m selected features.

Calculate $\boldsymbol{SNR}_i = \frac{\sum_{j=1}^{r} \hat{W}_{ij}^2}{\hat{\sigma}_i^2}, i \in \{1, 2, \cdots, d\}$

Sort the SNR values: $\boldsymbol{SNR}_{(1)} \leq \boldsymbol{SNR}_{(2)} \leq \cdots \leq \boldsymbol{SNR}_{(d)}$

Selected feature indices are: $\mathcal{I}_m = \{i : \boldsymbol{SNR}_i \geq \boldsymbol{SNR}_{(d-m+1)}\}$

2.4 Theoretical Guarantees

A central pursuit in modern high-dimensional statistics is to understand not only *if* an estimator converges to its true value, but at what *rate* it converges and with what level of certainty. Latent variable models, and in particular Latent Factor Analysis (LFA) and Probabilistic Principal Component Analysis (PPCA), represent a foundational toolkit for modeling and dimensionality reduction in complex datasets. For decades, the theoretical understanding of these models was primarily dominated by classical asymptotic results, which guarantee consistency and efficiency as the number of samples approaches infinity [37, 36, 6, 117].

However, while asymptotic results establish eventual consistency, they are insufficient for a rigorous analysis of an estimator's behavior in the non-limiting regime. A complete theoretical understanding requires finite-sample bounds. It provides explicit, high-probability guarantees on the estimation error for any given sample size n. Such bounds are not merely a practical refinement; they are a mathematical necessity for validating the downstream application of any estimator.

In the context of our work, the reliability of any feature selection procedure based on ranking estimated SNR values fundamentally depends on the fidelity of the SNR estimates themselves. A finite-sample bound establishes a high-probability "contract" that the estimated value $\boldsymbol{S\hat{N}R}$ lies within a quantifiable neighborhood of the actual, unobservable $\boldsymbol{SNR}$. This is critical, as it provides measurable confidence, ensuring that the estimated values used for any downstream task are not artifacts of sampling noise but are instead meaningful approximations of the true feature importance. Without such an explicit, non-asymptotic bound, the numbers produced by the estimator lack formal validation, making any subsequent analysis heuristic.

This thesis makes an initial step toward deriving such a bound. We have conducted a rigorous non-asymptotic analysis of the estimators for LFA and PPCA. We move beyond the traditional asymptotic regime to analyze the explicit nature of the parameter estimation errors for moderate values of n. Our theoretical framework is built upon the powerful tools of modern probability theory, leveraging seminal results in matrix concentration inequalities and the behavior of quadratic forms of sub-Gaussian random vectors [113, 126]. The primary contribution of this work is to translate these abstract mathematical tools into concrete, interpretable guarantees for the key parameters of latent factor models. The implications are significant: these bounds provide a formal basis for model validation, enable a deeper understanding of the statistical difficulty of parameter estimation

in the presence of noise, and offer a principled way to reason about the reliability of any downstream task that depends on these models.

Our analysis begins with bounding the deviation of the sample covariance matrix

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

from its population counterpart $\mathbf{\Sigma}$. This deviation directly governs the accuracy of estimated signal and noise variances in PPCA and LFA.

We present two complementary results: a *global bound* on the operator norm $\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}$, which controls the spectral deviation of all eigenvalues collectively, and a *local bound* on the diagonal elements $|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}|$, which quantifies feature-wise deviations.

Theorem 3 (due to [126]). *[Covariance Matrix Concentration] Assume that $M_1, \ldots, M_n$ are independent realizations of a $d \times d$ positive semi-definite symmetric random matrix M with mean $\mathbf{E}[M] = \mathbf{\Sigma}$. Let M satisfy for some $\kappa \geq 1$,*

$$\|\mathbf{x}^\top M \mathbf{x}\|_{\psi_1} \leq \kappa^2 \mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}, \quad \textit{for all } \mathbf{x} \in \mathbb{R}^d, \tag{2.15}$$

where $\|\cdot\|_{\psi_1}$ denotes the sub-exponential norm, defined as $\|\mathbf{Y}\|_{\psi_\alpha} = \inf\{c > 0 : \mathbf{E}[\exp(|\mathbf{Y}|^\alpha / c^\alpha)] \leq 2\}$. Then, for any $t > 0$, with probability at least $1 - \exp(-t)$, it holds that

$$\|\frac{1}{n}\sum_{i=1}^{n} M_i - \mathbf{\Sigma}\|_{op} \leq 20\kappa^2 \|\mathbf{\Sigma}\|_{op} \sqrt{\frac{4r(\mathbf{\Sigma}) + t}{n}}, \tag{2.16}$$

whenever $n \geq 4r(\mathbf{\Sigma}) + t$, where $r(\mathbf{\Sigma}) = \mathrm{tr}(\mathbf{\Sigma})/\|\mathbf{\Sigma}\|_{op}$ is the effective rank of $\mathbf{\Sigma}$.

For many latent variable models, the observed data vectors $\mathbf{x}_i$ are assumed to be zero-mean sub-Gaussian random vectors. In this case, we can set $M_i = \mathbf{x}_i \mathbf{x}_i^T$. For a centered Gaussian vector $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{\Sigma})$, the condition of Theorem 3 is met with $\kappa^2 = 8/3$. Considerable research [22, 3, 20, 63, 62] has focused on the problem of deriving non-asymptotic bounds for the operator norm error $\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}$. Theorem 3 is a special case of the deviation bound derived by [62], which provides a dimension(d)-free upper bound, ensuring the stability of eigenvalue-based estimators such as PPCA loadings. It implies that accurate signal recovery is possible whenever $n \gg r(\Sigma)$, even if $d \gg n$. The effective rank term $r(\Sigma)$ naturally emerges as a dimension-corrected complexity measure.

While the global bound provides a powerful worst-case guarantee on the overall deviation of the matrix in any direction, we also employ a tighter, more refined bound on the error of the individual

diagonal elements, $|\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}|$. If the feature variances are highly heterogeneous, the global bound can yield a loose bound for low-variance features. A tighter bound for individual diagonal elements, $|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}|$, can be obtained by applying a scalar concentration inequality directly.

Theorem 4 (Refined Bound for Sample Variance Error)**.** *Let $Z_k = x_{ki}^2 - \mathbb{E}[x_{ki}^2]$ be i.i.d. zero-mean random variables, where $x_{ki} \sim \mathcal{N}(0, \mathbf{\Sigma}_{ii})$. Let c be the universal constant from Bernstein's inequality. For any failure probability $\delta \in (0, 1)$, if the sample size n satisfies*

$$n > \frac{8}{c} \ln(4/\delta), \tag{2.17}$$

then the sample variance error is bounded with probability at least $1 - \delta/2$ by:

$$\left| \hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii} \right| \leq \sqrt{\frac{2}{c}} \mathbf{\Sigma}_{ii} \cdot \sqrt{\frac{\ln(4/\delta)}{n}}. \tag{2.18}$$

Proof. The term to be bounded is the average of n i.i.d. mean-zero random variables:

$$\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii} = \frac{1}{n} \sum_{k=1}^{n} \left(x_{ki}^2 - \mathbb{E}[x_{ki}^2] \right). \tag{2.19}$$

Let $Z_k = x_{ki}^2 - \mathbb{E}[x_{ki}^2]$. From the LFA model, x_{ki} is a zero-mean Gaussian variable with variance $\mathbf{\Sigma}_{ii} = (\boldsymbol{W}\boldsymbol{W}^T)_{ii} + \sigma_i^2$. Thus, Z_k is a centered, scaled chi-squared random variable ($\mathbf{\Sigma}_{ii}(\chi_1^2 - 1)$), which is sub-exponential.

To apply Bernstein's inequality[113], we identify its parameters. The variance parameter v is:

$$v = \mathrm{Var}(Z_k) = \mathrm{Var}(x_{ki}^2) = 2\mathbf{\Sigma}_{ii}^2. \tag{2.20}$$

The sub-exponential scale parameter b is proportional to the variance of the underlying Gaussian, so $b = C_b \mathbf{\Sigma}_{ii}$ for some universal constant C_b. For a χ_1^2 variable, a standard version of Bernstein's inequality uses parameters equivalent to $v = 2\mathbf{\Sigma}_{ii}^2$ and $b = 4\mathbf{\Sigma}_{ii}$.

Bernstein's inequality for the average provides the following bound on the tail probability:

$$\mathbb{P}\left(\left| \frac{1}{n} \sum_{k=1}^{n} Z_k \right| \geq \epsilon \right) \leq 2 \exp\left(-c \cdot n \cdot \min\left(\frac{\epsilon^2}{2\mathbf{\Sigma}_{ii}^2}, \frac{\epsilon}{4\mathbf{\Sigma}_{ii}} \right) \right), \tag{2.21}$$

where c is a universal constant. Setting the right-hand side to our desired failure probability $\delta/2$ and solving for ϵ gives:

$$\epsilon \geq \max\left(\sqrt{\frac{2\mathbf{\Sigma}_{ii}^2 \log(4/\delta)}{cn}}, \frac{4\mathbf{\Sigma}_{ii} \log(4/\delta)}{cn} \right). \tag{2.22}$$

This can be rewritten as:

$$\epsilon \geq \mathbf{\Sigma}_{ii} \cdot \max\left(\sqrt{\frac{2\log(4/\delta)}{cn}}, \frac{4\log(4/\delta)}{cn}\right). \tag{2.23}$$

The 'max' operator selects between the sub-Gaussian-like term (with $\sqrt{1/n}$) and the sub-exponential-like term (with $1/n$). The first term is larger if and only if:

$$\sqrt{\frac{2\log(4/\delta)}{cn}} > \frac{4\log(4/\delta)}{cn}. \tag{2.24}$$

Squaring both sides (which are positive) yields:

$$\frac{2\log(4/\delta)}{cn} > \frac{16(\log(4/\delta))^2}{c^2n^2}. \tag{2.25}$$

Assuming $\log(4/\delta) > 0$, we can simplify by multiplying by c^2n^2 and dividing by $cn\log(4/\delta)$:

$$2c > \frac{16\log(4/\delta)}{n}. \tag{2.26}$$

Rearranging gives the constraint on n:

$$n > \frac{8}{c}\log(4/\delta). \tag{2.27}$$

Under the constraint in Eq. (2.27), the first term in Eq. (2.23) dominates. The bound on the error simplifies to:

$$\left|\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}\right| \leq \mathbf{\Sigma}_{ii}\sqrt{\frac{2\log(4/\delta)}{cn}}. \tag{2.28}$$

□

This theorem ensures that the error in estimating the variance of a quiet, low-variance feature will not be artificially inflated by the presence of a noisy, high-variance feature elsewhere in the data. This local control will be the key to analyzing the parameter estimates of LFA.

2.4.1 PPCA

We now apply the concentration bounds to analyze the parameter estimates of the PPCA model. The following are the assumptions of our consideration.

(**A1**) Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be $d-$dimensional n i.i.d. samples from the PPCA model $\mathbf{x}_k = \mathbf{W}\boldsymbol{\gamma}_k + \boldsymbol{\epsilon}_k$.

(**A2**) $\boldsymbol{\gamma}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ and $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$. The random vectors $\boldsymbol{\gamma}_k$ and $\boldsymbol{\epsilon}_{k'}$ are independent for any $\{k, k' \in \{1, 2, \ldots n\}\}$.

This isotropic noise variance assumption ($\sigma^2\mathbf{I}_d$) enables a closed-form Maximum Likelihood (ML) solution based on an eigendecomposition of the sample covariance matrix $\hat{\mathbf{\Sigma}} = \mathbf{U}\mathbf{D}\mathbf{U}^T = \sum_{j=1}^{d} \lambda_j \mathbf{u}_j \mathbf{u}_j^T$. This estimation process has been described in (2.5). Our goal is to bound the estimation error of the principal subspace, represented by $\mathbf{W}\mathbf{W}^T$, by leveraging the bound on $|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}|_{ii}$.

To analyze the error in the estimated signal variance for a single feature, we first establish a key identity.

Lemma 1. *If the assumptions (**A1-A2**) hold, the i-th diagonal element of the PPCA signal estimate $(\widehat{\mathbf{W}\mathbf{W}^T})_{ii}$ can be written as:*

$$\begin{aligned} (\widehat{\mathbf{W}\mathbf{W}^T})_{ii} &= \hat{\mathbf{\Sigma}}_{ii} - \hat{\sigma}^2 - \Delta_i, \\ \left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| &\leq \left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|\hat{\sigma}^2 - \sigma^2 + \Delta_i\right|, \end{aligned} \tag{2.29}$$

where $(\hat{\mathbf{W}}, \hat{\sigma}^2)$ is defined in (2.5), l_j is the j^{th} eigenvalue and $\mathbf{u}_j$ is the j^{th} eigen vector of $\hat{\mathbf{\Sigma}}$, and $\Delta_i = \sum_{j=r+1}^{d} (l_j - \hat{\sigma}^2)(\mathbf{u}_j)_i^2$ is a remainder term.

Proof. The i-th diagonal element of $\hat{\mathbf{\Sigma}}$ is $\hat{\mathbf{\Sigma}}_{ii} = \sum_{j=1}^{d} l_j (\mathbf{u}_j)_i^2$. The eigenvectors form an orthonormal basis, so $\sum_{j=1}^{d} (\mathbf{u}_j)_i^2 = 1$. The estimated signal variance is $(\widehat{\mathbf{W}\mathbf{W}^T})_{ii} = \sum_{j=1}^{r} (l_j - \hat{\sigma}^2)(\mathbf{u}_j)_i^2$.

$$\begin{aligned} (\widehat{\mathbf{W}\mathbf{W}^T})_{ii} &= \sum_{j=1}^{r} l_j (\mathbf{u}_j)_i^2 - \hat{\sigma}^2 \sum_{j=1}^{r} (\mathbf{u}_j)_i^2 \\ &= \left(\hat{\mathbf{\Sigma}}_{ii} - \sum_{j=r+1}^{d} l_j (\mathbf{u}_j)_i^2 \right) - \hat{\sigma}^2 \left(1 - \sum_{j=r+1}^{d} (\mathbf{u}_j)_i^2 \right) \\ &= \hat{\mathbf{\Sigma}}_{ii} - \hat{\sigma}^2 - \left(\sum_{j=r+1}^{d} l_j (\mathbf{u}_j)_i^2 - \hat{\sigma}^2 \sum_{j=r+1}^{d} (\mathbf{u}_j)_i^2 \right) \\ &= \hat{\mathbf{\Sigma}}_{ii} - \hat{\sigma}^2 - \sum_{j=r+1}^{d} (l_j - \hat{\sigma}^2)(\mathbf{u}_j)_i^2. \end{aligned} \tag{2.30}$$

We have: $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T + \sigma^2 I_d$

Then, the difference is:

$$\begin{aligned} & |(\mathbf{W}\mathbf{W}^T)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii}| \\ &= |(\mathbf{\Sigma}_{ii} - \sigma^2) - (\hat{\mathbf{\Sigma}}_{ii} - \hat{\sigma}^2 - \Delta_i)| \\ &= |(\mathbf{\Sigma} - \hat{\mathbf{\Sigma}})_{ii} - (\sigma^2 - \hat{\sigma}^2) + \Delta_i| \\ &\leq |(\mathbf{\Sigma} - \hat{\mathbf{\Sigma}})_{ii}| + |\sigma^2 - \hat{\sigma}^2| + |\Delta_i| \end{aligned} \tag{2.31}$$

□

The isotropic noise variance assumption ($\sigma^2\mathbf{I}_d$) enables us to break down $\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right|$ in two parts as represented in (2.29). We have already showcased the bound for the first part (i.e. $\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right|$) in theorem 4. To bound the deviation in signal variance, we have to bound the second term in (2.29). The following theorem provides an element-wise bound for the error in estimating signal variances (i.e. $(\mathbf{W}\mathbf{W}^T)_{ii}$).

Theorem 5 (PPCA Element-wise Estimation Error)**.** *Let the assumptions of Theorem 4 hold along with the assumptions (**A1-A2**). For any $\delta \in (0,1)$, $n > \frac{8}{c}\log(4/\delta)$, with probability at least $1-\delta$:*

$$\begin{aligned}\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| &\leq \mathbf{\Sigma}_{ii}\sqrt{\frac{2\ln(4/\delta)}{cn}} \\ &\quad + C' \cdot \|\mathbf{\Sigma}\|_{op} \cdot \sqrt{\frac{r(\mathbf{\Sigma}) + \ln(2/\delta)}{n}} \\ &\leq \epsilon(\delta)\end{aligned} \tag{2.32}$$

where $\epsilon(\delta) = C \cdot \|\mathbf{\Sigma}\|_{op} \cdot \sqrt{\frac{r(\mathbf{\Sigma})+\ln(1/\delta)}{n}}$, $r(\mathbf{\Sigma}) = \frac{\mathrm{tr}(\mathbf{\Sigma})}{\|\mathbf{\Sigma}\|_{op}}$ and C is an constant.

Proof. We start from the error decomposition established previously in (2.29):

$$\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| \leq \left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|\hat{\sigma}^2 - \sigma^2 + \Delta_i\right|.$$

We bound each of the two terms using a union bound, allocating a probability of failure of $\delta/2$ to the first term and $\delta/2$ to the events driven by the operator norm.

Term 1: $\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right|$. This term represents the deviation of a single element of the sample covariance matrix. As established in 4, the average of the i.i.d. mean-zero sub-exponential variables $Z_k = (\mathbf{x}_k)_i^2 - \mathbf{\Sigma}_{ii}$ is bounded by applying Bernstein's inequality. For $n > \frac{8}{c}\log(4/\delta)$, with a failure probability of $\delta/2$, the bound is:

$$\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| \leq \mathbf{\Sigma}_{ii}\sqrt{\frac{2\log(4/\delta)}{cn}} \tag{2.33}$$

Term 2: We want to bound $|\hat{\sigma}^2 - \sigma^2 + \Delta_i|$.

$$\begin{aligned}\hat{\sigma}^2 - \sigma^2 + \Delta_i &= \left(\frac{1}{d-r}\sum_{j=r+1}^{d}(l_j - \sigma^2)\right) + \left(\sum_{j=r+1}^{d}(l_j - \hat{\sigma}^2)(u_j)_i^2\right) \\ &= \left(\frac{1}{d-r}\sum_{j=r+1}^{d}(l_j - \sigma^2)\right) + \left(\sum_{j=r+1}^{d}(l_j - \sigma^2 - (\hat{\sigma}^2 - \sigma^2))(u_j)_i^2\right) \\ &= (\hat{\sigma}^2 - \sigma^2) + \sum_{j=r+1}^{d}(l_j - \sigma^2)(u_j)_i^2 - (\hat{\sigma}^2 - \sigma^2)\sum_{j=r+1}^{d}(u_j)_i^2 \\ &= (\hat{\sigma}^2 - \sigma^2)\left(1 - \sum_{j=r+1}^{d}(u_j)_i^2\right) + \sum_{j=r+1}^{d}(l_j - \sigma^2)(u_j)_i^2. \quad (2.34)\end{aligned}$$

Using the identity $1 - \sum_{j=r+1}^{d}(u_j)_i^2 = \sum_{j=1}^{r}(u_j)_i^2$, from (2.34), we have:

$$= (\hat{\sigma}^2 - \sigma^2)\sum_{j=1}^{r}(u_j)_i^2 + \sum_{j=r+1}^{d}(l_j - \sigma^2)(u_j)_i^2.$$

Results like Weyl's[113] inequality relate the eigenvalues l_j of $\hat{\mathbf{\Sigma}}$ to the eigenvalues λ_j of the true covariance $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_d$. Specifically,

$$\max_{j=1,\ldots,d}|l_j - \lambda_j(\Sigma)| \leq \|S_n - \Sigma\|_{op}.$$

This helps control the error in the estimated eigenvalues $\hat{\lambda}_j$. Also note that, for PPCA, $\lambda_j = \sigma^2$, for $j \in \{r+1, r+2, \ldots, d\}$. Therefore, Using Weyl's inequality, we get:

$$\begin{aligned}|\hat{\sigma}^2 - \sigma^2| &= \left|\frac{1}{d-r}\sum_{j=r+1}^{d} l_j - \sigma^2\right| = \left|\frac{1}{d-r}\sum_{j=r+1}^{d}(l_j - \lambda_j)\right| \quad (\text{since } \lambda_j = \sigma^2 \text{ for } j > r) \\ &\leq \frac{1}{d-r}\sum_{j=r+1}^{d}|l_j - \lambda_j| \\ &\leq \max_{j=r+1,\ldots,d}|l_j - \lambda_j| \leq \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}\end{aligned} \quad (2.35)$$

Once again, we apply the triangle inequality and Weyl's inequality[113]:

$$\begin{aligned}|\hat{\sigma}^2 - \sigma^2 + \Delta_i| &\leq |\hat{\sigma}^2 - \sigma^2|\sum_{j=1}^{r}(u_j)_i^2 + \sum_{j=r+1}^{d}|l_j - \sigma^2|(u_j)_i^2 \\ &\leq \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}\sum_{j=1}^{r}(u_j)_i^2 + \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}\sum_{j=r+1}^{d}(u_j)_i^2 \quad (\text{due to}(2.35)) \\ &= \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}\left(\sum_{j=1}^{r}(u_j)_i^2 + \sum_{j=r+1}^{d}(u_j)_i^2\right) = \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}\sum_{j=1}^{d}(u_j)_i^2 = \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}. \quad (2.36)\end{aligned}$$

The bound for Term 2 depends on the operator norm bound $\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}$. With a probability of at least $1 - \delta/2$, we use the dimension-free bound from [126], which applies to sub-Gaussian vectors. To achieve a failure probability of $\delta' = \delta/2$,we set $\delta/2 = e^{-t}$, which implies $t = \log(2/\delta)$. Substituting this into the bound from that paper, we have that with probability at least $1 - \delta/2$:

$$\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op} \leq 20 \cdot \frac{8}{3} \cdot \|\mathbf{\Sigma}\|_{op} \cdot \sqrt{\frac{r(\mathbf{\Sigma}) + \log(2/\delta)}{n}}. \tag{2.37}$$

The bound for Term 1 also holds with probability at least $1 - \delta/2$ whenever $n > \frac{8}{c}\log(4/\delta)$.

Therefore with probability at least $1 - \delta$:

$$\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| \leq \mathbf{\Sigma}_{ii}\sqrt{\frac{2\log(4/\delta)}{cn}} + \frac{160}{3} \cdot \|\mathbf{\Sigma}\|_{op} \cdot \sqrt{\frac{r(\mathbf{\Sigma}) + \log(2/\delta)}{n}} \tag{2.38}$$

□

The non-asymptotic bound in Theorem 5 is composed of two distinct parts:

1. **A local bound for $\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right|$:** $2 \cdot \mathbf{\Sigma}_{ii}\sqrt{(\cdots)}$, which depends on the variance of the specific feature i.
2. **A global bound $\left\|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})\right\|_{op}$:** $\left\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\right\|_{op}$ arises, as we try to bound $\left|\hat{\sigma}^2 - \sigma^2 + \Delta_i\right|$: The operator bound appears as we apply Weyl's inequality from [113] to bound both $|\hat{\sigma}^2 - \sigma^2|$ and $|l_j - \sigma^2|$. As PPCA estimates the noise variance σ^2 by averaging information across all noise dimensions, it couples the estimation error of feature i to the behavior of all other features through the operator norm.

Before moving to the LFA error analysis, we present another lemma for the PPCA model to achieve a somewhat more local bound for $\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right|$.

Lemma 2. *Let the assumptions of Theorem 5 hold. For any $\delta \in (0, 1)$, when $n > \frac{8}{c}\log(4/\delta)$ with probability at least $1 - \delta$:*

$$\begin{aligned}\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| \leq \mathbf{\Sigma}_{ii}\sqrt{\frac{2\ln(4/\delta)}{cn}} \\ + \min\{\epsilon(\delta/2), 2\mathbf{\Sigma}_{ii}\}.\end{aligned} \tag{2.39}$$

Proof. From the previous Theorem 5

$$(\mathbf{W}\mathbf{W}^T)_{ii} = \mathbf{\Sigma}_{ii} - \sigma^2, \quad \text{as } \sigma^2 > 0, \text{therefore } (\mathbf{W}\mathbf{W}^T)_{ii} < \mathbf{\Sigma}_{ii} \tag{2.40}$$

$$(\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii} = \hat{\mathbf{\Sigma}}_{ii} - \left[\sum_{j=r+1}^{d} l_j(\mathbf{u}_j)_i^2 + \hat{\sigma}^2 \sum_{j=1}^{r} l_j(\mathbf{u}_j)_i^2\right] \quad (l_j \geq 0 \text{ for all } j \geq 1 \text{ as } \hat{\mathbf{\Sigma}} \text{ is psd}) \tag{2.41}$$

$$(\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii} < \hat{\mathbf{\Sigma}}_{ii} \tag{2.42}$$

By applying triangle inequality we get:

$$\begin{aligned}
&|\hat{\mathbf{W}}\hat{\mathbf{W}}^T - \mathbf{W}\mathbf{W}^T| \\
&\leq |\hat{\mathbf{\Sigma}}_{ii}| + |\mathbf{\Sigma}_{ii}| \\
&\leq |\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}| + 2|\mathbf{\Sigma}_{ii}| \qquad (2.43)
\end{aligned}$$

Previously we proved: $|\hat{\mathbf{W}}\hat{\mathbf{W}}^T - \mathbf{W}\mathbf{W}^T| \leq |\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}| + \|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}$

Combining the above two inequalities, we get: $|\hat{\mathbf{W}}\hat{\mathbf{W}}^T - \mathbf{W}\mathbf{W}^T| \leq |\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}| + \min\{\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{op}, 2\mathbf{\Sigma}_{ii}\}$

Therefore, the required bound can be obtained with probability at least $(1-\delta)$, by employing Theorems 4 and 3 □

Lemma 2 shows that for small n, the estimation error for i^{th} signal variance can be bounded locally by $\mathbf{\Sigma}_{ii}\sqrt{\frac{2\ln(4/\delta)}{cn}} + 2\mathbf{\Sigma}_{ii}$. It is a biased bound as $\mathbf{\Sigma}_{ii} \geq 0$ no matter how large the n value is. Therefore, as $n \to \infty$, more precisely when $n \geq \max\{\frac{C^2\|\mathbf{\Sigma}\|_{op}^2}{4\mathbf{\Sigma}_{ii}^2}(r(\mathbf{\Sigma}) + \ln(2/\delta)), \frac{8}{c}\log(4/\delta)\}$, the lemma uses the global bound for $\left\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\right\|_{op}$, which goes to 0 for large values of n.

Large d, large n. In this type of results, both the number of features d and the number of samples n tend to infinity, while the ratio $\frac{d}{n}(= \gamma \geq 0)$ is kept constant. This is also known as the "ultra-high dimensional" or "big data" regime. The main challenge to drawing inferences on asymptotic behaviors of eigenvalues and eigenvectors in this setup is that the sample covariance matrix does not well approximate the population covariance matrix unlike the case when d was fixed. There has been considerable effort to establish convergence results for sample eigenvalues and eigenvectors in recent years. Some of these findings will be discussed below.

Recent theoretical advancements on the 'large d large n' setup are based on the assumption that the data matrix, $\mathbf{X}$ is generated from a spiked population covariance model. The notion of the spiked population covariance model was first introduced by [58].

Definition 1. ***Spiked Covariance model.*** *Under this model, the data matrix $\boldsymbol{X}$, can be viewed as $\boldsymbol{X}^T = \boldsymbol{E}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Z}$, where $\boldsymbol{E} = [\boldsymbol{e_1}, \boldsymbol{e_2}, \cdots, \boldsymbol{e_d}]$ is a $d \times d$ orthogonal matrix, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and $\boldsymbol{Z}$ is a $d \times n$ matrix constructed with iid random variables $\boldsymbol{Z}_{ij}$ with $E(\boldsymbol{Z}_{ij}) = 0$, $E(\boldsymbol{Z}_{ij}^2) = 1$ and $E(\boldsymbol{Z}_{ij}^4) \leq \infty$. The population covariance matrix is $\boldsymbol{\Sigma} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^{\boldsymbol{T}}$. Here, λ_k's are assumed to follow a specific structure, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m > \lambda_{m+1} = \cdots = \lambda_d = 1$.*

The sample covariance matrix is $\hat{\boldsymbol{\Sigma}} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}/n = \boldsymbol{E}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{E}^T/n$.

The spectral decomposition of the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ is, $\hat{\boldsymbol{\Sigma}} = \boldsymbol{USU^T}$. Here, $\boldsymbol{S} = \text{diag}(s_1, s_2, \cdots s_d)$ are the ordered sample eigenvalues and $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \cdots, \boldsymbol{u_d}]$ is the corresponding $d \times d$ sample eigenvector matrix. For the remaining of the section, we assume that $\lim_{n\to\infty} \frac{d}{n} = \gamma$. Also there are k population eigenvalues such that $\lambda_i > 1 + \sqrt{\gamma}$, for $i \leq k$

The following result is due to [11].

Theorem 6. *[due to [11]] For $\gamma \in (0, 1)$, the following holds:*

$$s_i \overset{a.s.}{\to} \begin{cases} \rho(\lambda_i), & \text{if } i \leq k \\ (1 + \sqrt{\gamma})^2 & \text{otherwise,} \end{cases}$$

where $\rho(x) = x(1 + \frac{\gamma}{x-1})$.

It is evident from Theorem 6, the sample eigenvalues are not consistent estimates of the population counterparts. However, a consistent estimator can be found for $\lambda_i > 1 + \gamma$ using the following inverse function:

$$\rho^{-1}(x) = \frac{x + 1 - \gamma + \sqrt{(x + 1 - \gamma)^2 - 4x}}{2} \tag{2.44}$$

Also, it has been shown in [10] that s_i are asymptotically normal .

Although consistency could not be proved for $\gamma > 0$, [69] proved consistency for $\gamma = 0$.

Lemma 3 (due to [69])**.** *If $\lim_{n\to\infty} \frac{d}{n} = \gamma = 0$, then,*

$$s_i \overset{a.s.}{\to} \begin{cases} \lambda_i & \text{if } i \leq m \\ 1 & \text{otherwise.} \end{cases}$$

For eigenvectors, the convergence of the angle between sample eigen vectors($\mathbf{u}_i$) and population eigenvectors($\mathbf{e}_i$) has been proved for Gaussian $\boldsymbol{Z}_{ij}$'s in [84]. The author used the inner product between two unit vectors to represent the cosine angle between $\mathbf{u}_i$ and $\mathbf{e}_i$.

Theorem 7. *[due to [84]] Under the assumption of multiplicity one , if $\lim_{n\to\infty} \frac{d}{n} = \gamma \in (0, 1)$, and $\boldsymbol{Z_{ij}}$'s follow the standard normal distribution, then*

$$| < \boldsymbol{e}_i, \boldsymbol{u}_i > | \overset{a.s.}{\to} \begin{cases} \phi(\lambda_i) & \text{if } \lambda_i > 1 + \sqrt{\gamma} \\ 0 & \text{if } 1 < \lambda_i < 1 + \sqrt{\gamma}. \end{cases}$$

Here, $< \boldsymbol{a}, \boldsymbol{b} >$ represents the inner product between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\phi(x) = \sqrt{\frac{1 - \frac{\gamma}{(x-1)^2}}{1 + \frac{\gamma}{x-1}}}$.

[80] reached the same conclusion for $\gamma > 0$ using a matrix perturbation approach under the Gaussian random noise model. [69] generalized Theorem 7 by relaxing the distributional assumption and proved a weaker convergence (in probability) for the angles between population and sample eigenvectors, when $\gamma \geq 0$.

Lemma 4 (due to [69]). *Under the assumption of multiplicity one , if* $\lim_{n\to\infty} \frac{d}{n} = \gamma \geq 0$,

$$| < \boldsymbol{e}_i, \boldsymbol{u}_i > | \xrightarrow{p} \begin{cases} \phi(\lambda_i) & \text{if } \lambda_i > 1 + \sqrt{\gamma} \\ 0 & \text{if } 1 < \lambda_i < 1 + \sqrt{\gamma}. \end{cases}$$

2.4.2 LFA

We now turn to the more general Latent Factor Analysis (LFA) model, where the idiosyncratic noise $\boldsymbol{\Psi}$ is diagonal but not necessarily isotropic. This flexibility prevents a simple closed-form solution for the ML estimators, which are instead found using iterative algorithms such as Expectation-Maximization (EM). The details of the EM-process have been presented in Theorem 1. Our goal is to analyze the behavior of the estimation error for feature-specific parameters: the signal variances $(\mathbf{W}\mathbf{W}^T)_{ii}$ and the noise variances $\boldsymbol{\Psi}_{ii}$. One of the foundational asymptotic theories for the high-dimensional LFA problem was comprehensively established in the past by [9]. Their main contribution is the development of a complete asymptotic theory for the MLE of LFA parameters when $(n, d) \to \infty$. To achieve this goal, they have considered some assumptions on the model parameters:

(A3) The latent factors $\{\boldsymbol{\gamma}_i, i = 1, 2, \ldots, n\}$ are deterministic and non-random, with a sample covariance matrix converging to a positive definite matrix M_{ff}. This indicates that the factors are "strong" and stable.

(A4) The noise $\{\boldsymbol{\epsilon}_i, i = 1, 2, \ldots, n\}$ are iid random variables, also they are independent with $\{\boldsymbol{\gamma}_i, i = 1, 2, \ldots, n\}$. Also, $E(\boldsymbol{\epsilon}_i) = 0, E(\boldsymbol{\epsilon}_i^2) = \boldsymbol{\Psi}, E(\boldsymbol{\epsilon}_i^4) \leq C^4$, where $\boldsymbol{\Psi} = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots \sigma_d^2)$.

(A5) For some positive constant and $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, d\}$:

- $\|\mathbf{W}_{j\cdot}\|_2 \leq C$,
- $C^{-2} \leq \sigma^2 \leq C^2$, where $\text{var}(\boldsymbol{\epsilon}_i)_{jj} = \sigma_j^2$,
- $\mathbf{W}^T \boldsymbol{\Psi}^{-1} \mathbf{W}/d \to \mathbf{C}$,
- $lim_{d\to\infty} \frac{1}{d} \sum_{j=1}^{d} \sigma_j^{-4} ((\mathbf{W}_{j\cdot} \otimes \mathbf{W}_{j\cdot})(\mathbf{W}_{j\cdot}^T \otimes \mathbf{W}_{j\cdot}^T)) = \Omega$,

where $(\mathbf{C}, \Omega)$ are positive definite matrices. This ensures that influential, detectable factors are present in high dimensions.

(A6) The model parameters (specifically σ_j^2) are estimated to lie within a compact set (C^{-2}, C^2), a standard technical requirement for proving the consistency of complex, non-linear estimators.

The latent factor models are generally non-identifiable without additional constraints. Therefore, authors have introduced additional constraints (**IC1-IC5**) in [9] to ensure the full identifiability of the model parameters. Now we will state the theorem that proves the consistency of the estimators and establishes their rates of convergence in the high high-dimensional data regime:

Theorem 8 (due to [9]). *Under Assumptions (**A3-A6**), when $d, n \to \infty$, with any one of the identification conditions(**IC1-IC5**) mentioned in [9], we have:*

$$\frac{1}{d}\sum_{i=1}^{d}\frac{1}{\hat{\sigma}_i^2}\|\hat{\mathbf{W}}_{i,\cdot} - \mathbf{W}_{i,\cdot}\|^2 = O_p(\frac{1}{n}), \tag{2.45}$$

$$\frac{1}{d}\sum_{i=}^{d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 = O_p(\frac{1}{n}), \tag{2.46}$$

where, $X_n = O_p\left(\frac{1}{n}\right)$ means for any $\epsilon > 0$, there exists a finite, positive constant M such that for all $n \geq 1$: $P\left(|nX_n| > M\right) < \epsilon$.

The results of [9] provide a clear idea of the convergence rates for $\hat{\mathbf{W}}$ and $\hat{\mathbf{\Psi}}$. The four assumptions, defined in (**A3-A6**), collectively ensure the factor model is well-posed for high-dimensional analysis and guarantee that a stable underlying factor signal can be consistently estimated because it remains statistically detectable amidst well-behaved, feature-specific noise as the dimensions of the data grow to infinity.

Specifically, the assumption **A5** plays a significant role in ensuring this stability. It also tells us how the SNRs will behave for high-dimensional data. In our work SNRs are defined as:

$$\boldsymbol{SNR}_i = \frac{1}{\sigma_i^2}\sum_{j=1}^{r}\mathbf{W}_{ij}^2, i \in \{1, 2, \cdots, d\}. \tag{2.47}$$

The condition in (**A5**), the fact that the limit $im_{d\to\infty}\frac{1}{d}\sum_{i=1}^{d}\sigma_i^{-4}((\mathbf{W}_{i\cdot}\otimes\mathbf{W}_{i\cdot}))(\mathbf{W}_{i\cdot}\otimes\mathbf{W}_{i\cdot}) = \Omega$, a fixed matrix, implies the following:

- The contributions of the features must, on average, be well-behaved. This prevents pathological scenarios. For instance, it implies that the signal strengths $\{\sum_{j=1}^{r}\mathbf{W}_{ij}^2, i \in \{1, \ldots, n\}$ cannot be growing in a wild, unbounded way relative to the noise variances as we add more features. If the SNRs were systematically exploding or behaving too erratically, this sum would not converge to a stable limit.

- The weighting term σ_i^{-4} is the strongest link to the SNR. Features with low idiosyncratic noise (σ_i^2) are weighted extremely heavily in this sum. These are the features that are likely to have high SNRs. Therefore, the assumption can be rephrased more intuitively: The long-term stability and learnability of the entire factor model is determined by the collective properties of its most informative (highest-SNR) features. The noisy, low-SNR features contribute very little to the sum and are effectively ignored.

- The condition that the resulting matrix Ω must be positive definite is a statement about non-redundancy. For example, this condition would be violated if 90% of the high-SNR features

were strongly related to Factor 1 but had almost no relation to Factors 2 through r. In such a case, we would learn a lot about Factor 1, but the information about the other factors would be weak, and the matrix Ω would become singular (not positive definite).

They have also shown that the same set of estimates has limiting distributions (i.e., asymptotic normality). The entirety of their results — consistency, rates, and distributions — is asymptotic i.e. when both d and n approach infinity. These results do not provide a quantifiable bound on the estimation error for any fixed, finite sample size n. Their $O_p(1/n)$ rate tells us about the scaling in the limit, but not the constants or higher-order terms that govern performance for a real-world n.

Now, we will prove the following lemma, which is a direct implication of Theorem 8:

Lemma 5. *Under the assumptions **A3-A6**, as* $(d, n) \to \inf$, $\max_{i=1,\dots,d} |\hat{\sigma}_i^2 - \sigma_i^2| = O_P\left(\sqrt{\frac{d}{n}}\right)$.

Proof. Let $\hat{\mathbf{\Psi}} = \text{diag}(\hat{\sigma}_1^2, \dots, \hat{\sigma}_d^2)$ be the estimated noise variance matrix. From Theorem 5.1 of [9],

$$\frac{1}{d}\sum_{i=1}^{d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 = O_P\left(\frac{1}{n}\right). \tag{2.48}$$

From this, we can derive the Frobenius norm bound for the difference between $\hat{\mathbf{\Psi}}$ and $\mathbf{\Psi}$:

$$\|\hat{\mathbf{\Psi}} - \mathbf{\Psi}\|_F^2 = \sum_{i=1}^{d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 = d \cdot \left(\frac{1}{d}\sum_{i=1}^{d}(\hat{\sigma}_i^2 - \sigma_i^2)^2\right) = d \cdot O_P\left(\frac{1}{n}\right) = O_P\left(\frac{d}{n}\right).$$

Taking the square root, we get:

$$\|\hat{\mathbf{\Psi}} - \mathbf{\Psi}\|_F = O_P\left(\sqrt{\frac{d}{n}}\right). \tag{2.49}$$

From the Frobenius norm bound, we can directly derive a bound for the maximum element-wise deviation. Since the Frobenius norm is the square root of the sum of squared elements, the largest squared element must be less than or equal to the sum of all squared elements:

$$\max_{i=1,\dots,d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 \leq \sum_{i=1}^{d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 = \|\hat{\mathbf{\Psi}} - \mathbf{\Psi}\|_F^2.$$

Substituting the Frobenius norm bound:

$$\max_{i=1,\dots,d}(\hat{\sigma}_i^2 - \sigma_i^2)^2 = O_P\left(\frac{d}{n}\right).$$

Taking the square root, we get the bound for the maximum absolute deviation:

$$\max_{i=1,\dots,d}|\hat{\sigma}_i^2 - \sigma_i^2| = O_P\left(\sqrt{\frac{d}{n}}\right). \tag{2.50}$$

□

However, the constraints (**IC1-IC5**) to ensure identifiability have only been applied to prove asymptotic results for $|\hat{\mathbf{W}} - \mathbf{W}|$, not for $|\hat{\sigma}_i^2 - \sigma_i^2|$; therefore, the constraints (**IC1-IC5**) have also not been considered in Lemma 5.

However, moving from asymptotic convergence to explicit, finite-sample bounds of order $f(d, n)$ presents a formidable theoretical challenge. The difficulty stems directly from the inherent structure of the LFA estimation problem. Unlike models with closed-form solutions, where errors can be propagated directly, LFA parameters are the output of an iterative procedure, such as the Expectation-Maximization (EM) algorithm [92]. This iterative process establishes a profound and intricate relationship between the estimates of the signal loadings $\hat{\mathbf{W}}$ and the noise variances $\hat{\mathbf{\Psi}}$. As a result, a simple error analysis is intractable. As we proceed, we will discuss the challenges one faces when deriving tight bounds for the ML estimates of latent factor model parameters. For our analysis, we will also consider the following assumptions:

(**A7**) $\{\mathbf{x}_i\}_{i=1}^n$ be n i.i.d. samples from a d-dimensional Latent Factor Analysis (LFA) model: $\mathbf{x}_i = \mathbf{W}\boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi})$,

(**A8**) $\boldsymbol{\gamma}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ and $\mathbf{\Psi} = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$. The random vectors $\boldsymbol{\gamma}_k$ and $\boldsymbol{\epsilon}_{k'}$ are independent for any $\{k \neq k' \in \{1, 2, \ldots n\}\}$

(**A9**) The true, underlying parameters of the LFA model are assumed to be well-behaved:

- (i) $\|\mathbf{W}_{i\cdot}\|_2 \leq C_W$,
- (ii) $0 < \sigma_{\min}^2 \leq \sigma_j^2 \leq \sigma_{\max}^2$ for all $j \in \{1, \ldots, d\}$.

(**A10**) The parameter estimates $(\hat{\mathbf{W}}, \hat{\mathbf{\Psi}})$ produced by the estimation procedure in Theorem 1 are assumed to be regular in the sense that their corresponding operator norms are bounded.

- (i) $\|\hat{\mathbf{W}}_{i\cdot}\|_2 \leq C_{\hat{W}}$,
- (ii) $\|\hat{\boldsymbol{\beta}}_{i\cdot}\|_2 \leq C_\beta$, where $\hat{\boldsymbol{\beta}} = \hat{\mathbf{W}}^T(\hat{\mathbf{W}}\hat{\mathbf{W}}^T + \hat{\mathbf{\Psi}})^{-1}$.

Now we prove a set of essential equations that hold when the EM algorithm for LFA converges.

Theorem 9 (Stationary Point Characterization of the LFA Log-Likelihood)**.** *The EM algorithm for LFA converges to a point where the parameter estimates* $(\hat{\mathbf{W}}, \hat{\mathbf{\Psi}})$ *satisfy certain fixed-point equations, as characterized in Theorem 1. Suppose Assumptions* (***A7,A8,A9(ii) and A6***) *hold. At a stationary point of the LFA log-likelihood (see [92]), the following two identities hold:*

$$(\mathbf{I} - \mathbf{M}^{-1}\widehat{\mathbf{\Sigma}})\mathbf{M}^{-1}\hat{\mathbf{W}} = \mathbf{0}, \tag{2.51}$$

$$D(\mathbf{M}^{-1}\widehat{\mathbf{\Sigma}}) = \mathbf{I}, \tag{2.52}$$

where $\mathbf{M} = \hat{\mathbf{W}}\hat{\mathbf{W}}^\top + \hat{\mathbf{\Psi}}$*, and* $D(\mathbf{X})$ *denotes the diagonal matrix formed by the diagonal elements of* $\mathbf{X}$*.*

Furthermore, the following identity holds for each coordinate $i \in \{1, \ldots, d\}$*:*

$$(\hat{\mathbf{W}}\hat{\mathbf{W}}^\top)_{ii} = \widehat{\mathbf{\Sigma}}_{ii} - \hat{\mathbf{\Psi}}_{ii}. \tag{2.53}$$

Consequently, we obtain the following inequalities:

$$\left|(\mathbf{W}\mathbf{W}^\top)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^\top)_{ii}\right| \leq \left|(\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right|, \tag{2.54}$$

$$\left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right| \leq \left|(\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|(\mathbf{W}\mathbf{W}^\top)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^\top)_{ii}\right|. \tag{2.55}$$

Proof. The M-step (of the EM algorithm) update equation for $\mathbf{W}$ is given in Theorem 1, with converged $\mathbf{W}$):

$$\hat{\mathbf{W}} = \left(\sum_{i=1}^{n} \mathbf{x}_i E(\boldsymbol{\gamma}_i|\mathbf{x}_i)^T\right)\left(\sum_{i=1}^{n} E(\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T|\mathbf{x}_i)\right)^{-1}$$

Let's simplify the terms:

- **Numerator sum:** Substituting $E(\boldsymbol{\gamma}_i|\mathbf{x}_i) = \boldsymbol{\beta}\mathbf{x}_i$:

$$\begin{aligned}\sum_{i=1}^{n} \mathbf{x}_i E(\boldsymbol{\gamma}_i|\mathbf{x}_i)^T &= \sum_{i=1}^{n} \mathbf{x}_i(\boldsymbol{\beta}\mathbf{x}_i)^T = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}^T \\ &= \left(\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T\right)\boldsymbol{\beta}^T = (n\hat{\mathbf{\Sigma}})\boldsymbol{\beta}^T.\end{aligned}$$

- **Denominator sum:** Substituting $E(\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T|\mathbf{x}_i) = \mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}^T$:

$$\begin{aligned}\sum_{i=1}^{n} E(\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T|\mathbf{x}_i) &= \sum_{i=1}^{n}(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}^T) \\ &= n(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}}) + \boldsymbol{\beta}\left(\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T\right)\boldsymbol{\beta}^T \\ &= n(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}}) + \boldsymbol{\beta}(n\hat{\mathbf{\Sigma}})\beta^T \\ &= n(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\hat{\mathbf{\Sigma}}\beta^T).\end{aligned}$$

Now, substitute these simplified sums back into the $\hat{\mathbf{W}}$ equation:

$$\begin{aligned}\hat{\mathbf{W}} &= (n\hat{\mathbf{\Sigma}})\boldsymbol{\beta}^T(n(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\hat{\mathbf{\Sigma}}\beta^T))^{-1} \\ &= (n\hat{\mathbf{\Sigma}})\boldsymbol{\beta}^T\frac{1}{n}(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\hat{\mathbf{\Sigma}}\boldsymbol{\beta}^T)^{-1} \\ &= \hat{\mathbf{\Sigma}}\boldsymbol{\beta}^T(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \boldsymbol{\beta}\hat{\mathbf{\Sigma}}\boldsymbol{\beta}^T)^{-1}.\end{aligned}$$

Multiplying both sides by $(\mathbf{I}_r - \beta\hat{\mathbf{W}} + \boldsymbol{\beta}\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^T)$ from the right, we obtain:

$$\hat{\mathbf{W}}(\mathbf{I}_r - \boldsymbol{\beta}\hat{\mathbf{W}} + \beta\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^T) = \hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^T.$$

To further apply matrix algebra, we substitute $\boldsymbol{\beta} = \hat{\mathbf{W}}^T(\hat{\boldsymbol{\Psi}} + \hat{\mathbf{W}}\hat{\mathbf{W}}^T)^{-1}$. Let $\mathbf{M} = \hat{\boldsymbol{\Psi}} + \hat{\mathbf{W}}\hat{\mathbf{W}}^T$, so $\boldsymbol{\beta} = \hat{\mathbf{W}}^T\mathbf{M}^{-1}$. Also note that $\hat{\mathbf{W}}\hat{\mathbf{W}}^T = \mathbf{M} - \hat{\boldsymbol{\Psi}}$.

The identity becomes:

$$\begin{aligned}
\hat{\mathbf{W}} - \hat{\mathbf{W}}\boldsymbol{\beta}\hat{\mathbf{W}} + \hat{\mathbf{W}}\boldsymbol{\beta}\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^T &= \hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^T, \\
\hat{\mathbf{W}} - \hat{\mathbf{W}}(\hat{\mathbf{W}}^T\mathbf{M}^{-1})\hat{\mathbf{W}} + \hat{\mathbf{W}}(\hat{\mathbf{W}}^T\mathbf{M}^{-1})\hat{\boldsymbol{\Sigma}}(\mathbf{M}^{-1}\hat{\mathbf{W}}) &= \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}}, \\
\hat{\mathbf{W}} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)\mathbf{M}^{-1}\hat{\mathbf{W}} + (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}} &= \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}}, \\
\hat{\mathbf{W}} - (\mathbf{M} - \hat{\boldsymbol{\Psi}})\mathbf{M}^{-1}\hat{\mathbf{W}} + (\mathbf{M} - \hat{\boldsymbol{\Psi}})\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}} &= \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}}, \\
\hat{\mathbf{W}} - (I - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1})\hat{\mathbf{W}} + (I - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1})\hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}} &= \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}}, \\
\hat{\mathbf{W}} - \hat{\mathbf{W}} + \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\mathbf{W}} + \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}} - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}} &= \hat{\boldsymbol{\Sigma}}\mathbf{M}^{-1}\hat{\mathbf{W}}.
\end{aligned}$$

Simplifying, moving all terms to one side and factoring out $\mathbf{M}^{-1}\hat{\mathbf{W}}$, we obtain:

$$(\hat{\boldsymbol{\Psi}} + \hat{\boldsymbol{\Sigma}} - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}} - \hat{\boldsymbol{\Sigma}})\mathbf{M}^{-1}\hat{\mathbf{W}} = \mathbf{1}.$$

This simplifies to:

$$(\hat{\boldsymbol{\Psi}} - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}})\mathbf{M}^{-1}\hat{\mathbf{W}} = \mathbf{1},$$

$$\hat{\boldsymbol{\Psi}}(\mathbf{I} - \mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}})\mathbf{M}^{-1}\hat{\mathbf{W}} = \mathbf{1}.$$

Assuming $\hat{\boldsymbol{\Psi}}$(**A9(ii),A6**) is invertible, we get

$$\boxed{(\mathbf{I} - \mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}})\mathbf{M}^{-1}\hat{\mathbf{W}} = \mathbf{1}}. \tag{2.56}$$

This identity must hold at the maximum likelihood estimate for $\hat{\mathbf{W}}$ and $\hat{\boldsymbol{\Psi}}$.

Next, let us discuss the identity for $\hat{\boldsymbol{\Psi}}$. The M-step update equation for $\hat{\boldsymbol{\Psi}}$ is given as in Theorem 1, with converged $\hat{\boldsymbol{\Psi}}$:

$$\hat{\boldsymbol{\Psi}} = \frac{1}{n}D\left(\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T - \hat{\mathbf{W}}E(\boldsymbol{\gamma}_i|\mathbf{x}_i)\mathbf{x}_i^T\right) \tag{2.57}$$

Let's simplify the sum term inside $D(\cdot)$:

$$\begin{aligned}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T - \hat{\mathbf{W}}E(\boldsymbol{\gamma}_i|\mathbf{x}_i)\mathbf{x}_i^T &= \sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T - \hat{\mathbf{W}}\sum_{i=1}^{n}E(\boldsymbol{\gamma}_i|\mathbf{x}_i)\mathbf{x}_i^T \\ &= n\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\sum_{i=1}^{n}(\boldsymbol{\beta}\mathbf{x}_i)\mathbf{x}_i^T \\ &= n\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\boldsymbol{\beta}\left(\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T\right) \\ &= n\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\boldsymbol{\beta}(n\hat{\boldsymbol{\Sigma}}).\end{aligned}$$

Substitute this back into the $\hat{\boldsymbol{\Psi}}$ equation:

$$\begin{aligned}\hat{\boldsymbol{\Psi}} &= \frac{1}{n}D(n\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\boldsymbol{\beta}(n\hat{\boldsymbol{\Sigma}})) \\ &= D(\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\boldsymbol{\beta}\hat{\boldsymbol{\Sigma}}).\end{aligned}$$

Now, substitute $\boldsymbol{\beta} = \hat{\mathbf{W}}^T\mathbf{M}^{-1}$:

$$\hat{\boldsymbol{\Psi}} = D(\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}(\hat{\mathbf{W}}^T\mathbf{M}^{-1})\hat{\boldsymbol{\Sigma}}),$$

$$\hat{\boldsymbol{\Psi}} = D(\hat{\boldsymbol{\Sigma}} - \hat{\mathbf{W}}\hat{\mathbf{W}}^T\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}),$$

Substitute $\hat{\mathbf{W}}\hat{\mathbf{W}}^T = \mathbf{M} - \hat{\boldsymbol{\Psi}}$:

$$\begin{aligned}\hat{\boldsymbol{\Psi}} &= D(\hat{\boldsymbol{\Sigma}} - (\mathbf{M} - \hat{\boldsymbol{\Psi}})\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}) \\ &= D(\hat{\boldsymbol{\Sigma}} - (\mathbf{I} - \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1})\hat{\boldsymbol{\Sigma}}) \\ &= D(\hat{\boldsymbol{\Sigma}} - \hat{\boldsymbol{\Sigma}} + \hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}).\end{aligned}$$

The $\hat{\boldsymbol{\Sigma}}$ terms cancel out:

$$\hat{\boldsymbol{\Psi}} = D(\hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}).$$

This is a critical identity. Given that $\hat{\boldsymbol{\Psi}}$ is a diagonal matrix, we can state that its i-th diagonal element is equal to the i-th diagonal element of $\hat{\boldsymbol{\Psi}}\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}$.

If $\hat{\boldsymbol{\Psi}}$ is invertible (**A6**), we can imply a further identity by "dividing" by $\hat{\boldsymbol{\Psi}}$ (element-wise on the diagonal operation):

$$\boxed{\mathbf{I} = D(\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}})}. \tag{2.58}$$

This identity states that the diagonal elements of the product $\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}$ must be equal to 1. Since $\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}}$ is symmetric, this is equivalent to saying that $(\mathbf{M}^{-1}\hat{\boldsymbol{\Sigma}})_{ii} = 1$ for all $i = 1, \ldots, d$.

Now we will proceed to prove the next part of the theorem. Let $\mathbf{A} = \hat{\mathbf{\Sigma}} - \mathbf{M}$. Then $\hat{\mathbf{\Sigma}} = \mathbf{M} + \mathbf{A}$ and $\mathbf{M}^{-1}\hat{\mathbf{\Sigma}} = \mathbf{I} + \mathbf{M}^{-1}\mathbf{A}$ so Eq. (2.56) becomes

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{M}^{-1}\mathbf{W} = 0,$$

which after multiplying to the left with $\mathbf{M}$ becomes

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{W} = 0.$$

Multiply by $\mathbf{W}^T$ to the right and we obtain

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{W}\mathbf{W}^T = \mathbf{A}\mathbf{M}^{-1}(\mathbf{M} - \mathbf{\Psi}) = 0.$$

which means

$$\mathbf{A} = \mathbf{A}\mathbf{M}^{-1}\mathbf{\Psi}.$$

The identity from (2.58) becomes

$$(\mathbf{M}^{-1}\mathbf{A})_{ii} = 0.$$

But $\mathbf{A} = \mathbf{A}^T$ so $(\mathbf{A}\mathbf{M}^{-1})_{ii} = 0$. But in this case, since $\mathbf{A} = \mathbf{A}\mathbf{M}^{-1}\mathbf{\Psi}$, we also have that $\mathbf{A}_{ii} = 0$ for all i.

This implies that:

$$\boxed{(\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii} = \hat{\mathbf{\Sigma}}_{ii} - \hat{\mathbf{\Psi}}_{ii}.} \tag{2.59}$$

Let us proceed to the last part of the theorem.

$$\begin{aligned} \left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii}\right| &= \left|(\mathbf{\Sigma_{ii}} - \mathbf{\Psi}_{ii}) - (\hat{\mathbf{\Sigma}}_{ii} - \hat{\mathbf{\Psi}}_{ii})\right| \\ &= \left|(\mathbf{\Sigma_{ii}} - \hat{\mathbf{\Sigma}}_{\mathbf{ii}}) + (\hat{\mathbf{\Psi}}_{\mathbf{ii}} - \mathbf{\Psi_{ii}})\right| \\ &\leq \left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right|. \end{aligned} \tag{2.60}$$

□

This theorem is powerful because it allows us to decompose the error in the signal variance estimate. The problem is now reduced to bounding the sample variance error (which we can do with Theorem 4) and the noise variance estimation error. However, the derivation for the finite sample bound for $(\mathbf{W}\mathbf{W}^\top, \mathbf{\Psi})$ is complicated and computation heavy.

Before that, we will derive some asymptotic bounds for signals and SNR estimation error in the following theorem.

Theorem 10. *Under the assumption **A5-A8**, When $(n, d) \to \infty$, then*

$$\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii}\right| = O_p(\sqrt{d/n}) \tag{2.61}$$

$$\left|\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i\right| = O_p(\sqrt{d/n}) \tag{2.62}$$

Proof. From Theorem 9, we get: $\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii}\right| \leq \left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| + \left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right|$.

From Theorem 4, we get $\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right| = O_p(1/\sqrt{n})$ and from Lemma 5, it can be derived that $\left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right| = O_p(\sqrt{d/n})$.

Therefore, $\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^T)_{ii}\right| = O_p(\max(1/\sqrt{n}, \sqrt{d/n})) = O_p(\sqrt{d/n})$

Now we, turn to SNRs. Let X_0, Y_0 be true values and X, Y be estimators. If $1/c \leq Y \leq c$, for some $c > 0$ then,

$|(XY_0 - X_0Y)/(YY_0)| = |(X - X_0)Y_0 - X_0(Y - Y_0)| / (|Y|Y_0) \leq \frac{c}{Y_0}(Y_0|X - X_0| + X_0|Y - Y_0|)$.

We wish to bound the error $\left|\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i\right|$. Note that, $\boldsymbol{SNR}_i = \frac{\mathbf{W}\mathbf{W}^T_{ii}}{\sigma_i^2} = \frac{\mathbf{\Sigma}_{ii} - \sigma_i^2}{\sigma_i^2} = \frac{\mathbf{\Sigma}_{ii}}{\sigma_i^2} - 1$. Therefore, the numerator and denominator terms are:

- True Numerator: $X_0 = \mathbf{\Sigma}_{ii}$
- Estimated Numerator: $X = \hat{\mathbf{\Sigma}}_{ii}$
- True Denominator: $Y_0 = \sigma_i^2$
- Estimated Denominator: $Y = \hat{\sigma}_i^2$

Assumption **A6** assures that there is a c, such that $1/c \leq \sigma_i^2 \leq c$ for sufficiently large n. Therefore, we get the following:

$$\begin{aligned}
\left|\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i\right| &= \left|\frac{X}{Y} - \frac{X_0}{Y_0}\right| \\
&\leq \frac{c}{Y_0}\left(Y_0\left|X - X_0\right| + X_0\left|Y - Y_0\right|\right) \\
&= c\left(\left|\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}\right| + \boldsymbol{SNR}_i\left|\hat{\sigma}_i^2 - \sigma_i^2\right|\right) \qquad (2.63) \\
&= c(O_p(\sqrt{1/n}) + \boldsymbol{SNR}_i \times O_p(\sqrt{d/n})) = O_p(\sqrt{d/n}). \qquad (2.64)
\end{aligned}$$

□

Here we present the three remarks, which provide an upper bound for $\left|(\mathbf{W}\mathbf{W}^\top)_{ii} - (\hat{\mathbf{W}}\hat{\mathbf{W}}^\top)_{ii}\right|$, $\left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right|$, and $\left|\hat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i\right|$.

Remark 1. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be $d-$dimensional n i.i.d. samples from the LFA model $\mathbf{x}_k = \mathbf{W}\boldsymbol{\gamma}_k + \boldsymbol{\epsilon}_k$. Suppose Assumptions (**A5-A8**, **A10**) hold.*

For any $\delta \in (0,1)$, if $n > \ln(6/\delta)$, then with probability at least $1-\delta$, the noise variance estimation error is bounded by:

$$|\tilde{\sigma}_i^2 - \sigma_i^2| \leq K \cdot \sqrt{\frac{\ln(6/\delta)}{n}} + \mu_i,$$

and when, $n \geq \max\{\frac{8}{c}\ln(4/\delta), \ln(12/\delta)\}$, with probability at least $1-\delta$, the signal variance estimation error is bounded by:

$$\left|(\mathbf{W}\mathbf{W}^T)_{ii} - (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}\right| \leq \left(\boldsymbol{\Sigma}_{ii}\sqrt{2/c} + K\right)\sqrt{\frac{\ln(c'/\delta)}{n}} + \mu_i \tag{2.65}$$

where μ_i is defined as:

$$\mu_i = (\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{\Psi})_{ii} \tag{2.66}$$

and c' is an universal constant. Also,

$$\mu_i \leq B_i, i \in \{1, \ldots, n\}$$

and d is fixed and $\mu_i \to 0$ as $n \to \infty$. Here, the constant B_i is defined in the proof and depends on the assumed constants $C_W, C_{\hat{W}}, C_\beta$, and $\sigma^2_{\max}$.

Proof. The proof strategy is to decompose the error term into a sum of averages of i.i.d. mean-zero random variables and then apply the appropriate concentration inequalities to each term using a union bound.

Error Decomposition. The error is decomposed into three main terms:

$$\hat{\sigma}_i^2 - \sigma_i^2 = \frac{1}{n}\sum_{k=1}^{n}\left[\epsilon_{ki}^2 + (\mathbf{W}\gamma_k)_i^2 + 2(\mathbf{W}\gamma_k)_i\epsilon_{ki} - (\hat{\mathbf{W}}\hat{\gamma}_k)_i((\mathbf{W}\gamma_k)_i + \epsilon_{ki}) - \sigma_i^2\right].$$

Rearrange:

$$\hat{\sigma}_i^2 - \sigma_i^2 = \frac{1}{n}\sum_{k=1}^{n}\left(\epsilon_{ki}^2 - \sigma_i^2\right) + \frac{1}{n}\sum_{k=1}^{n}\left[(\mathbf{W}\gamma_k)_i^2 - (\hat{\mathbf{W}}\hat{\gamma}_k)_i(\mathbf{W}\gamma_k)_i + 2(\mathbf{W}\gamma_k)_i\epsilon_{ki} - (\hat{\mathbf{W}}\hat{\gamma}_k)_i\epsilon_{ki}\right].$$

Rewrite the second term:

$$(\mathbf{W}\gamma_k)_i^2 - (\hat{\mathbf{W}}\hat{\gamma}_k)_i(\mathbf{W}\gamma_k)_i = (\mathbf{W}\gamma_k)_i\left[(\mathbf{W}\gamma_k)_i - (\hat{\mathbf{W}}\hat{\gamma}_k)_i\right],$$

$$2(\mathbf{W}^*\gamma_k)_i\epsilon_{ki} - (\mathbf{W}_{\text{new}}\hat{\gamma}_k)_i\epsilon_{ki} = \epsilon_{ki}\left[2(\mathbf{W}^*\gamma_k)_i - (\mathbf{W}_{\text{new}}\hat{\gamma}_k)_i\right].$$

Define $u_{ki} = (\mathbf{W}\gamma_k)_i - (\hat{\mathbf{W}}\hat{\gamma}_k)_i$. Then:

$$2(\mathbf{W}\gamma_k)_i - (\hat{\mathbf{W}}\hat{\gamma}_k)_i = (\mathbf{W}\gamma_k)_i + \left[(\mathbf{W}\gamma_k)_i - (\hat{\mathbf{W}}\hat{\gamma}_k)_i\right] = (\mathbf{W}\gamma_k)_i + u_{ki}.$$

Thus:

$$\hat{\sigma}_i^2 - \sigma_i^2 = \underbrace{\frac{1}{n}\sum_{k=1}^{n}\left(\epsilon_{ki}^2 - \sigma_i^2\right)}_{\text{Term 1}} + \underbrace{\frac{1}{n}\sum_{k=1}^{n}[(\mathbf{W}\boldsymbol{\gamma}_k)_i \mathbf{u}_{ki}]}_{\text{Term 2a}} + \underbrace{\frac{1}{n}\sum_{k=1}^{n}[\epsilon_{ki}((\mathbf{W}\boldsymbol{\gamma}_k)_i + \mathbf{u}_{ki})]}_{\text{Term 2b}}, \tag{2.67}$$

where $\mathbf{u}_{ki} = (\mathbf{W}\boldsymbol{\gamma}_k)_i - (\hat{\mathbf{W}}\hat{\boldsymbol{\gamma}}_k)_i$. We allocate a failure probability of $\delta/3$ to each term.

Term 1 (Noise Variance Error). This is the average of i.i.d. mean-zero variables $Z_k = \epsilon_{ki}^2 - \sigma_i^2$. Since $\epsilon_{ki} \sim \mathcal{N}(0, \sigma_i^2)$, Z_k is a scaled centered chi-squared variable, which is sub-exponential with parameters $(v, b) = (2\sigma_i^2, 4\sigma_i^2)$. Applying the two-sided Bernstein's inequality ([113], Eq. 2.20), with probability at least $1 - \delta/3$:

$$\left|\frac{1}{n}\sum_{k=1}^{n}(\epsilon_{ki}^2 - \sigma_i^2)\right| \leq K_{1i} \cdot \max\left(\sqrt{\frac{\log(6/\delta)}{n}}, \frac{\log(6/\delta)}{n}\right). \tag{2.68}$$

Where $K_{1i} = c\sigma_i^2$, c is the constant from Bernstein's inequality.

Term 2a (Signal-Signal Cross-Term)

We begin by defining Term 2a as the average of n i.i.d. random variables T_k:

$$\text{Term 2a} = \frac{1}{n}\sum_{k=1}^{n} T_k, \tag{2.69}$$

where T_k is given by

$$T_k = (\boldsymbol{W}\boldsymbol{\gamma}_k)_i \cdot \left[(\boldsymbol{W}\boldsymbol{\gamma}_k)_i - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\gamma}}_k)_i\right]. \tag{2.70}$$

In general, the expectation of T_k is non-zero. Let $\mu_{2a} = \mathbb{E}[T_k]$. The error is bounded using the triangle inequality:

$$|\text{Term 2a}| \leq |\mu_{2a}| + \left|\frac{1}{n}\sum_{k=1}^{n}(T_k - \mu_{2a})\right|. \tag{2.71}$$

We will bound the deterministic bias term $|\mu_{2a}|$ and the zero-mean fluctuation term separately.

Term 2a Bias Term. The expectation μ_{2a} is computed as:

$$\mu_{2a} = \mathbb{E}[T_k] = (\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii}. \tag{2.72}$$

We bound its magnitude using the triangle inequality and properties of matrix norms:

$$|\mu_{2a}| \leq |(\boldsymbol{W}\boldsymbol{W}^T)_{ii}| + |(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii}|. \tag{2.73}$$

Under the assumptions, we get the operator norms , the terms are bounded as follows:

$$|(\boldsymbol{W}\boldsymbol{W}^T)_{ii}| = \|\boldsymbol{W}\|_2^2 \leq C_W^2, \tag{2.74}$$

$$|(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii}| \leq \|\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T\|_2 \leq \|\hat{\boldsymbol{W}}\|_2\|\hat{\boldsymbol{\beta}}\|_2\|\boldsymbol{W}\|_2^2 \leq C_{\hat{W}}C_\beta C_W^2. \tag{2.75}$$

Combining these gives the bound on the bias:

$$|\mu_{2a}| \leq (C_W^2 + C_{\hat{W}}C_\beta C_W^2) = O(C_W^2 + C_{\hat{W}}C_\beta C_W^2) =: K_{2a,\text{bias}}. \tag{2.76}$$

This constant is a polynomial in the assumed regularity constants and is independent of n and δ.

Term 2a Fluctuation Term. To bound the fluctuation term, we express the centered summands $\tilde{T}_k = T_k - \mu_{2a}$ as a centered quadratic form and apply the Hanson-Wright inequality. First, define the concatenated Gaussian vector $\boldsymbol{z}_k \in \mathbb{R}^{r+d}$:

$$\boldsymbol{z}_k = \begin{pmatrix} \boldsymbol{\gamma}_k \\ \boldsymbol{\epsilon}_k \end{pmatrix} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}), \tag{2.77}$$

where the covariance matrix $\boldsymbol{C}$ is

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi} \end{pmatrix}. \tag{2.78}$$

The operator norm of $\boldsymbol{C}$ is bounded by $\|\boldsymbol{C}\|_{\text{op}} = \max(1, \sigma_{\max}^2) =: \sigma_C$. We define two deterministic row vectors, $\boldsymbol{a}_i^\top$ and $\boldsymbol{b}_i^\top$:

$$\boldsymbol{a}_i^\top = \begin{pmatrix} \boldsymbol{W}_{i,:} & \boldsymbol{0}_{1\times d} \end{pmatrix} \in \mathbb{R}^{1\times(r+d)}, \tag{2.79}$$

$$\boldsymbol{b}_i^\top = \begin{pmatrix} \boldsymbol{W}_{i,:} - (\hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}})\boldsymbol{W} & -\hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}} \end{pmatrix} \in \mathbb{R}^{1\times(r+d)}. \tag{2.80}$$

With these, the components of T_k are linear forms of $\boldsymbol{z}_k$, and T_k is a quadratic form:

$$T_k = (\boldsymbol{a}_i^\top \boldsymbol{z}_k)(\boldsymbol{b}_i^\top \boldsymbol{z}_k) = \boldsymbol{z}_k^\top(\boldsymbol{a}_i\boldsymbol{b}_i^\top)\boldsymbol{z}_k. \tag{2.81}$$

We use the symmetric part of the matrix, defining $\boldsymbol{A}_i$:

$$\boldsymbol{A}_i = \frac{1}{2}\left(\boldsymbol{a}_i\boldsymbol{b}_i^\top + \boldsymbol{b}_i\boldsymbol{a}_i^\top\right). \tag{2.82}$$

Now, $T_k = \boldsymbol{z}_k^\top \boldsymbol{A}_i \boldsymbol{z}_k$, and its expectation is $\mathbb{E}[T_k] = \text{tr}(\boldsymbol{A}_i\boldsymbol{C}) = \mu_{2a}$. The centered variable is:

$$\tilde{T}_k = T_k - \mu_{2a} = \boldsymbol{z}_k^\top \boldsymbol{A}_i \boldsymbol{z}_k - \text{tr}(\boldsymbol{A}_i\boldsymbol{C}). \tag{2.83}$$

The Hanson-Wright inequality states that for a sum of such i.i.d. centered variables,

$$\mathbb{P}\left(\left|\sum_{k=1}^n \tilde{T}_k\right| \geq t\right) \leq 2\exp\left(-c\min\left(\frac{t^2}{n\|\boldsymbol{C}^{1/2}\boldsymbol{A}_i\boldsymbol{C}^{1/2}\|_F^2}, \frac{t}{\|\boldsymbol{C}^{1/2}\boldsymbol{A}_i\boldsymbol{C}^{1/2}\|_{\text{op}}}\right)\right), \tag{2.84}$$

for some universal constant $c > 0$. Let $\boldsymbol{M}_i = \boldsymbol{C}^{1/2}\boldsymbol{A}_i\boldsymbol{C}^{1/2}$. We need to bound the norms of $\boldsymbol{M}_i$.

The norms of $\boldsymbol{M}_i$ depend on the norms of $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$.

$$\|\boldsymbol{a}_i\|_2 = \|\boldsymbol{W}_{i,:}\|_2 \leq \|\boldsymbol{W}\|_2 \leq C_{WR}. \tag{2.85}$$

For $\boldsymbol{b}_i$, we have:

$$\begin{aligned}
\|\boldsymbol{b}_i\|_2^2 &= \|\boldsymbol{W}_{i,:} - \hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}}\boldsymbol{W}\|_2^2 + \|\hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}}\|_2^2 && (2.86)\\
&\leq \left(\|\boldsymbol{W}_{i,:}\|_2 + \|\hat{\boldsymbol{W}}_{i,:}\|_2\|\hat{\boldsymbol{\beta}}\|_{\text{op}}\|\boldsymbol{W}\|_{\text{op}}\right)^2 + \left(\|\hat{\boldsymbol{W}}_{i,:}\|_2\|\hat{\boldsymbol{\beta}}\|_{\text{op}}\right)^2 && (2.87)\\
&\leq \left(C_W + C_{\hat{W}}C_\beta C_W\right)^2 + \left(C_{\hat{W}}C_\beta\right)^2. && (2.88)
\end{aligned}$$

Let C_b be the square root of the right-hand side, which is a polynomial in the constants.

$$\|\boldsymbol{b}_i\|_2 \leq C_b. \tag{2.89}$$

The norms of $\boldsymbol{A}_i$ are bounded by:

$$\begin{aligned}
\|\boldsymbol{A}_i\|_{\text{op}} &\leq \|\boldsymbol{a}_i\|_2\|\boldsymbol{b}_i\|_2 \leq C_W C_b = O(C_W C_\beta C_W C_{\hat{W}}), && (2.90)\\
\|\boldsymbol{A}_i\|_F &\leq \sqrt{2}\|\boldsymbol{a}_i\|_2\|\boldsymbol{b}_i\|_2 \leq \sqrt{2}C_W C_b. && (2.91)
\end{aligned}$$

Finally, we bound the norms of $\boldsymbol{M}_i$:

$$\begin{aligned}
\|\boldsymbol{M}_i\|_{\text{op}} &\leq \|\boldsymbol{C}\|_{\text{op}}\|\boldsymbol{A}_i\|_{\text{op}} \leq \sigma_C C_W C_b =: K_{A,\text{op}}, && (2.92)\\
\|\boldsymbol{M}_i\|_F &\leq \|\boldsymbol{C}\|_{\text{op}}\|\boldsymbol{A}_i\|_F \leq \sigma_C\sqrt{2}C_W C_b =: K_{AF}. && (2.93)
\end{aligned}$$

We now solve for the bound on the average fluctuation. Let $\epsilon = t/n$. For a failure probability of $\delta/3$, the Hanson-Wright inequality is:

$$\frac{\delta}{3} \geq 2\exp\left(-c \cdot n \cdot \min\left(\frac{\epsilon^2}{K_{AF}^2}, \frac{\epsilon}{K_{A,\text{op}}}\right)\right). \tag{2.94}$$

Solving for ϵ yields two conditions that must be met:

$$\begin{aligned}
\epsilon &\geq \frac{K_{AF}}{\sqrt{c}}\sqrt{\frac{\log(6/\delta)}{n}}, && (2.95)\\
\epsilon &\geq \frac{K_{A,\text{op}}}{c}\frac{\log(6/\delta)}{n}. && (2.96)
\end{aligned}$$

To satisfy both, ϵ must be at least the maximum of the two lower bounds. We define a constant $K_{2a,\text{fluc}}$:

$$K_{2a,\text{fluc}} := \max\left(\frac{K_{AF}}{\sqrt{c}}, \frac{K_{A,\text{op}}}{c}\right). \tag{2.97}$$

This gives the high-probability bound on the fluctuation term:

$$\left|\frac{1}{n}\sum_{k=1}^{n}(T_k - \mu_{2a})\right| \leq K_{2a,\text{fluc}} \max\left(\sqrt{\frac{\log(6/\delta)}{n}}, \frac{\log(6/\delta)}{n}\right). \tag{2.98}$$

Final Bound for Term 2a. By combining the bounds for the bias and the fluctuation, we arrive at the final bound for Term 2a, which holds with probability at least $1 - \delta/3$:

$$|\text{Term 2a}| \leq K_{2a,\text{bias}} + K_{2a,\text{fluc}} \max\left(\sqrt{\frac{\log(6/\delta)}{n}}, \frac{\log(6/\delta)}{n}\right). \tag{2.99}$$

Term 2b (Signal-Noise Cross-Term)

We define Term 2b as the average of n i.i.d. random variables Y_k:

$$\text{Term 2b} = \frac{1}{n}\sum_{k=1}^{n} Y_k, \tag{2.100}$$

where Y_k is given by

$$Y_k = \epsilon_{ki}\left((\boldsymbol{W}\boldsymbol{\gamma}_k)_i + \boldsymbol{u}_{ki}\right) = \epsilon_{ki}\left(2(\boldsymbol{W}\boldsymbol{\gamma}_k)_i - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\gamma}}_k)_i\right). \tag{2.101}$$

The expectation of Y_k is generally non-zero. Let $\mu_{2b} = \mathbb{E}[Y_k]$. We bound the error using the triangle inequality:

$$|\text{Term 2b}| \leq |\mu_{2b}| + \left|\frac{1}{n}\sum_{k=1}^{n}(Y_k - \mu_{2b})\right|. \tag{2.102}$$

We proceed by bounding the deterministic bias $|\mu_{2b}|$ and then the zero-mean fluctuation term.

Term 2b Bias Term The expectation μ_{2b} is computed by taking the expectation over $\boldsymbol{\gamma}_k$ and $\boldsymbol{\epsilon}_k$. The terms involving products of independent zero-mean variables vanish:

$$\begin{aligned}
\mu_{2b} = \mathbb{E}[Y_k] &= \mathbb{E}\left[\epsilon_{ki}\left(2(\boldsymbol{W}\boldsymbol{\gamma}_k)_i - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}(\boldsymbol{W}\boldsymbol{\gamma}_k + \boldsymbol{\epsilon}_k))_i\right)\right] && (2.103)\\
&= \mathbb{E}\left[2\epsilon_{ki}(\boldsymbol{W}\boldsymbol{\gamma}_k)_i\right] - \mathbb{E}\left[\epsilon_{ki}(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{\gamma}_k)_i\right] - \mathbb{E}\left[\epsilon_{ki}(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{\epsilon}_k)_i\right] && (2.104)\\
&= 0 - 0 - \mathbb{E}\left[\epsilon_{ki}\sum_{j=1}^{d}(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}})_{ij}\epsilon_{kj}\right]. && (2.105)
\end{aligned}$$

Since $\boldsymbol{\Psi}$ is diagonal, $\mathbb{E}[\epsilon_{ki}\epsilon_{kj}] = \sigma_i^2$ if $j = i$ and 0 otherwise. This simplifies to:

$$\mu_{2b} = -(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}})_{ii}\sigma_i^2. \tag{2.106}$$

We bound its magnitude using the regularity assumptions:

$$\begin{aligned}
|\mu_{2b}| &= |(\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}})_{ii}|\sigma_i^2\\
&\leq \|\hat{\boldsymbol{W}}\|_2\|\hat{\boldsymbol{\beta}}\|_2\sigma_{\max}^2 \leq C_{\hat{W}}C_\beta\sigma_{\max}^2 =: K_{2b,\text{bias}}.
\end{aligned} \tag{2.107}$$

Term 2b Fluctuation Term. We express the centered summands $\tilde{Y}_k = Y_k - \mu_{2b}$ as a centered quadratic form of the concatenated Gaussian vector $\boldsymbol{z}_k = (\boldsymbol{\gamma}_k^\top, \boldsymbol{\epsilon}_k^\top)^\top \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C})$. The variable Y_k is a product of two linear forms, $Y_k = (\boldsymbol{c}_i^\top \boldsymbol{z}_k)(\boldsymbol{d}_i^\top \boldsymbol{z}_k)$. The first form represents ϵ_{ki}:

$$\boldsymbol{c}_i^\top = \begin{pmatrix} \mathbf{0}_{1\times r} & \boldsymbol{e}_i^\top \end{pmatrix} \in \mathbb{R}^{1\times(r+d)}, \tag{2.108}$$

where $\boldsymbol{e}_i$ is the i-th standard basis vector in $\mathbb{R}^d$. The second form represents $2(\boldsymbol{W}\boldsymbol{\gamma}_k)_i - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\gamma}}_k)_i$:

$$\boldsymbol{d}_i^\top = \begin{pmatrix} 2\boldsymbol{W}_{i,:} - (\hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}})\boldsymbol{W} & -\hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}} \end{pmatrix} \in \mathbb{R}^{1\times(r+d)}. \tag{2.109}$$

Thus, Y_k is the quadratic form:

$$Y_k = \boldsymbol{z}_k^\top (\boldsymbol{c}_i \boldsymbol{d}_i^\top) \boldsymbol{z}_k. \tag{2.110}$$

We use the symmetric matrix $\boldsymbol{B}_i$:

$$\boldsymbol{B}_i = \frac{1}{2}\left(\boldsymbol{c}_i \boldsymbol{d}_i^\top + \boldsymbol{d}_i \boldsymbol{c}_i^\top\right). \tag{2.111}$$

Now, $Y_k = \boldsymbol{z}_k^\top \boldsymbol{B}_i \boldsymbol{z}_k$, and its expectation is $\mathbb{E}[Y_k] = \operatorname{tr}(\boldsymbol{B}_i \boldsymbol{C}) = \mu_{2b}$. The centered variable is:

$$\tilde{Y}_k = Y_k - \mu_{2b} = \boldsymbol{z}_k^\top \boldsymbol{B}_i \boldsymbol{z}_k - \operatorname{tr}(\boldsymbol{B}_i \boldsymbol{C}). \tag{2.112}$$

We apply the Hanson-Wright inequality to the sum $\sum_{k=1}^n \tilde{Y}_k$, which requires bounding the norms of the matrix $\boldsymbol{N}_i = \boldsymbol{C}^{1/2} \boldsymbol{B}_i \boldsymbol{C}^{1/2}$.

The norms of $\boldsymbol{N}_i$ depend on the norms of $\boldsymbol{c}_i$ and $\boldsymbol{d}_i$.

$$\|\boldsymbol{c}_i\|_2 = \|\boldsymbol{e}_i\|_2 = 1. \tag{2.113}$$

For $\boldsymbol{d}_i$, we have:

$$\begin{aligned} \|\boldsymbol{d}_i\|_2^2 &= \|2\boldsymbol{W}_{i,:} - \hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}}\boldsymbol{W}\|_2^2 + \| - \hat{\boldsymbol{W}}_{i,:}\hat{\boldsymbol{\beta}}\|_2^2 && (2.114)\\ &\le \left(2\|\boldsymbol{W}_{i,:}\|_2 + \|\hat{\boldsymbol{W}}_{i,:}\|_2 \|\hat{\boldsymbol{\beta}}\|_{\mathrm{op}} \|\boldsymbol{W}\|_{\mathrm{op}}\right)^2 + \left(\|\hat{\boldsymbol{W}}_{i,:}\|_2 \|\hat{\boldsymbol{\beta}}\|_{\mathrm{op}}\right)^2 && (2.115)\\ &\le \left(2C_W + C_{\hat{W}} C_\beta C_W\right)^2 + \left(C_{\hat{W}} C_\beta\right)^2 = O(C_{\hat{W}}^2 C_\beta^2 C_W^2) && (2.116) \end{aligned}$$

We assume that $C_{\hat{W}}^2 C_\beta^2 C_W^2$ is much larger than C_W^2. Let C_d be the square root of the right-hand side, which is a polynomial in the constants.

$$\|\boldsymbol{d}_i\|_2 \le C_d. \tag{2.117}$$

The norms of $\boldsymbol{B}_i$ are bounded by:

$$\|\boldsymbol{B}_i\|_{\text{op}} \leq \|\boldsymbol{c}_i\|_2 \|\boldsymbol{d}_i\|_2 \leq C_d, \tag{2.118}$$

$$\|\boldsymbol{B}_i\|_F \leq \sqrt{2}\|\boldsymbol{c}_i\|_2 \|\boldsymbol{d}_i\|_2 \leq \sqrt{2}C_d. \tag{2.119}$$

Finally, we bound the norms of $\boldsymbol{N}_i = \boldsymbol{C}^{1/2}\boldsymbol{B}_i\boldsymbol{C}^{1/2}$:

$$\|\boldsymbol{N}_i\|_{\text{op}} \leq \|\boldsymbol{C}\|_{\text{op}} \|\boldsymbol{B}_i\|_{\text{op}} \leq \sigma_C C_d =: K_{B,\text{op}}, \tag{2.120}$$

$$\|\boldsymbol{N}_i\|_F \leq \|\boldsymbol{C}\|_{\text{op}} \|\boldsymbol{B}_i\|_F \leq \sigma_C \sqrt{2} C_d =: K_{BF}. \tag{2.121}$$

Final Bound for Term 2b. By combining the bounds for the bias and the fluctuation, we arrive at the final bound for Term 2b, which holds with probability at least $1 - \delta/3$:

$$|\text{Term 2b}| \leq K^i_{2b,\text{bias}} + K_{2b,\text{fluc}} \max\left(\sqrt{\frac{\log(6/\delta)}{n}}, \frac{\log(6/\delta)}{n}\right). \tag{2.122}$$

Where,

$$K_{2b,\text{fluc}} := \max\left(\frac{K_{BF}}{\sqrt{c}}, \frac{K_{B,\text{op}}}{c}\right). \tag{2.123}$$

Final Combination. By the union bound, all three bounds((2.68),(2.99),(2.122)) hold simultaneously with probability at least $1 - \delta$. Summing the bounds, we get:

$$|\tilde{\sigma}_i^2 - \sigma_i^2| \leq K_i \cdot \max\left(\sqrt{\frac{\log(6/\delta)}{n}}, \frac{\log(6/\delta)}{n}\right) + K_{2b,\text{bias}} + K_{2a,\text{bias}}. \tag{2.124}$$

For $n > \log(6/\delta)$,

$$\boxed{|\tilde{\sigma}_i^2 - \sigma_i^2| \leq K_i \cdot \sqrt{\frac{\log(6/\delta)}{n}} + \mu_i}. \tag{2.125}$$

Defining the final constant $K_i := K_{1i} + K_{2a,\text{fluc}} + K_{2b,\text{fluc}}$ and $\mu_i = (\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{\Psi})_{ii}$. From the bias term analysis we get: for $i \in \{1, \ldots, n\}$, $\mu_i \leq K_{2b,\text{bias}} + K_{2a,\text{bias}} = B_i$.

When d is fixed and $n \to \infty$, we get ML estimates $(\hat{\mathbf{W}}, \hat{\beta})$ which are consistent estimators of $\mathbf{W}, \beta$ (i.e. $(\hat{\mathbf{W}}, \hat{\beta}) \xrightarrow{p} (\mathbf{W}, \beta)$). Therefore, for sufficiently large n, if we replace $(\hat{\mathbf{W}}, \hat{\beta})$ with $(\mathbf{W}, \beta)$, we get the following:

$$\begin{aligned}
\mu^i_{2a} + \mu^i_{2b} &= (\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}\boldsymbol{\Psi})_{ii} \\
&= (\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\boldsymbol{W}\boldsymbol{\beta}\boldsymbol{W}\boldsymbol{W}^T)_{ii} - (\boldsymbol{W}\boldsymbol{\beta}\boldsymbol{\Psi})_{ii} \\
&= (\boldsymbol{W}\boldsymbol{W}^T(\mathbf{I_d} - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\Sigma} - \boldsymbol{\Psi})) - (\boldsymbol{W}\boldsymbol{\beta}\boldsymbol{\Psi}))_{ii} \\
&= (\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi} - \boldsymbol{W}\boldsymbol{\beta}\boldsymbol{\Psi})_{ii} \\
&= (\boldsymbol{W}\boldsymbol{\beta}\boldsymbol{\Psi} - \boldsymbol{W}\boldsymbol{\beta}\boldsymbol{\Psi})_{ii} = 0,
\end{aligned} \tag{2.126}$$

which completes the proof of the noise estimation error bound.

Now we will prove the next part. The result follows directly from the error decomposition in Equation (2.60). We bound the two terms on the right-hand side separately.

The first term, $\left|(\hat{\mathbf{\Sigma}} - \mathbf{\Sigma})_{ii}\right|$, is the element-wise covariance error. By Theorem 4, this is bounded by $\mathbf{\Sigma}_{ii}\sqrt{2\log(4/\delta)/cn}$ with probability at least $1 - \delta/2$, whenever $n \geq \frac{8}{c}\log(4/\delta)$.

The second term, $\left|\hat{\mathbf{\Psi}}_{ii} - \mathbf{\Psi}_{ii}\right|$, is the noise variance estimation error. By Theorem 1, this is bounded by $K_i \cdot \sqrt{\frac{\log(12/\delta)}{n}} + B_i$ with probability at least $1 - \delta/2$, whenever $n > \log(12/\delta)$.

Applying a union bound to combine these two events and absorbing constants gives the final result. □

Now we will present our following remark, which will connect the estimation error associated with $(\mathbf{\Sigma}_{ii}, \sigma_i^2)$ with the estimation error of SNR_i.

Remark 2 (SNR Estimation Error Bound)**.** *For the i-th feature in the LFA model, let the true SNR be* $SNR_i = (\mathbf{W}\mathbf{W}^T)_{ii}/\sigma_i^2$ *and its estimate be* $\widehat{SNR}_i = (\widehat{\mathbf{W}\mathbf{W}^T})_{ii}/\hat{\sigma}_i^2$*. Let the assumptions of Remark 1 hold.*

If the sample size n satisfies $n \geq \frac{2K_i}{(\sigma_i^2 - 2B_i)^2}\ln(\frac{12}{\delta})$*, then with probability at least* $1 - \delta$*, the SNR estimation error is bounded by:*

$$|\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i| \leq \frac{2c}{\sigma_i^2}\mathbf{\Sigma}_{ii} \cdot \sqrt{\frac{\ln(4/\delta)}{n}} + \frac{2\boldsymbol{SNR}_i}{(\sigma_i^2)}(K \cdot \sqrt{\frac{\log(12/\delta)}{n}} + B_i) \tag{2.127}$$

where (K, B_i) *are defined in Theorem 1.*

Proof. From theorem 10, we observe that bound of $|\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i|$, depends upon $|\sigma_i^2 - \hat{\sigma}_i^2|$. We assume that, $\left|\hat{\sigma}_i^2 - \sigma_i^2\right| \leq \sigma_i^2/2$ Therefore, we require the sample size n to be large enough so that this bound is less than or equal to $\sigma_i^2/2$:

$$K \cdot \sqrt{\frac{\log(12/\delta)}{n}} + B_i \leq \frac{\sigma_i^2}{2} \implies n \geq \frac{2K_i\log(\frac{12}{\delta})}{(\sigma_i^2 - 2B_i)^2}).$$

This is the sample size condition stated in the theorem. Assuming this condition holds, we can proceed, we want the error bound hold with probability $(1 - \frac{\delta}{2})$

Note $\left|\hat{\sigma}_i^2 - \sigma_i^2\right| \leq \sigma_i^2/2 \implies |\widehat{\boldsymbol{SNR}}_i - \boldsymbol{SNR}_i| \leq \frac{2}{(\sigma_i^2)}\left|\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}\right| + \frac{2SNR_i}{(\sigma_i^2)}\left|\hat{\sigma}_i^2 - \sigma_i^2\right|$ Now we will get the required bound by replacing the $\left|\hat{\mathbf{\Sigma}}_{ii} - \mathbf{\Sigma}_{ii}\right|$ and $\left|\hat{\sigma}_i^2 - \sigma_i^2\right|$, with their finite sample bound, derived in Theorem 4 and Remark 1 □

Discussion of the SNR Estimation Error Bound and its Implications. The implications of this result can be deconstructed into several key points.

First and foremost, the theorem provides the critical transition **from an asymptotic promise to a finite-sample guarantee**. The convergence guarantees in our previous work assured us that with enough data, we would eventually identify the correct features. This result quantifies that process, providing a non-asymptotic error bound that holds for any given n. The explicit dependence on n via the $1/\sqrt{n}$ term establishes the rate of convergence, confirming that the estimator behaves as expected, with the statistical error diminishing at the standard parametric rate. This allows a user to understand how the precision of their SNR estimates will improve with the collection of more data.

Second, the structure of the bound is deeply informative. It is composed of two distinct parts: a **systematic bias term** (B) and a **statistical fluctuation term** that scales with $1/\sqrt{n}$. The statistical term represents the random error from finite sampling. The bias term, however, is a finite sample correction. It represents an error component that *does vanish* as the number of samples n increases. This bias arises from using the estimated parameters themselves within the iterative EM estimation procedure, coupling the estimates in a way that introduces a persistent, systematic deviation. Our analysis makes this bias explicit, demonstrating that while the LFA-derived SNR is a statistically stable estimate, it is not, in general, an unbiased one for finite n. This is a critical piece of knowledge for anyone using LFA for precise quantitative modeling.

Finally, and most importantly, this theorem provides the **formal justification for our feature selection methodology**. It proves that the SNR we compute from data is not an arbitrary, noisy value but a statistically stable quantity that is explicitly and controllably close to the true SNR. This guarantee is what allows us to confidently rank features based on their estimated SNR values, knowing that this ranking is a meaningful reflection of the features' true, underlying importance. It elevates our method from a successful heuristic to a theoretically grounded and provably reliable engineering solution.

2.5 Simulations

We test the efficiency of previously discussed latent factor models through simulations. The methods employ dimension reduction through feature selection, and these simulations enable the measurement of the accuracy in recovering the true features. This is tested over varying sample sizes n and noise levels d_{noise}.

2.5.1 Simulation Procedure

The simulated sample has the following form:

$$\mathbf{X}_{n\times d} = \left(\mathbf{X}^{(1)}_{n\times 10}, \mathbf{X}^{(2)}_{n\times d_{noise}}\right) \tag{2.128}$$

Here, the d-dimensional observation vector $\mathbf{X}$ is a concatenation of relevant $\mathbf{X}^{(1)}$ and noisy (irrelevant) $\mathbf{X}^{(2)}$ dimensions. In this simulation setup, the number of relevant features (i.e., the dimension of $\mathbf{X}^{(1)}$) is fixed at 10, and we have experimented with different numbers of noisy dimensions, d_{noise}. The simulation procedure is described in detail in the following steps:

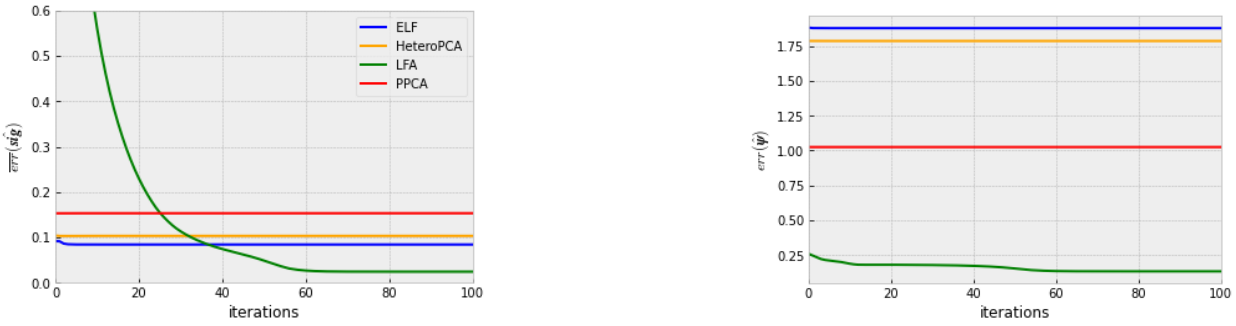
1. We assume the SNR values corresponding to $\mathbf{X}^{(1)}$ are positive. It is intuitive to generate $\mathbf{X}^{(1)}$ based on a pre-fixed set of positive SNR values. For a given set of signals, smaller SNRs correspond to large error variances. To make the true feature recovery more challenging, we choose SNRs to range from 0.5 (small) to 1.4 (large): $\boldsymbol{SNR}^*[i] = (15-i)/10, i \in \{1, ..., 10\}$.
2. Generate the coefficient matrix $\mathbf{W}$ associated with $\boldsymbol{\gamma}_i$s and used for computing signals.
   (a) $\mathbf{W}_{10\times r} = [\mathbf{W}_{ij}]$, $\mathbf{W}_{ij} \overset{\text{iid}}{\sim} N(0,1)$
3. Generate the error vector $\mathbf{e}^{(1)} \in \mathbb{R}^{10}$, $\mathbf{e}^{(1)}_i \overset{\text{iid}}{\sim} N(\mathbf{0}, \boldsymbol{\Psi}^{(1)*})$. Here $\boldsymbol{\Psi}^{(1)^*} = \text{diag}(\sigma^{*2}_1, \sigma^{*2}_2, \cdots, \sigma^{*2}_{10})$ and $\sigma^{*2}_j = \frac{\sum_{l=1}^{r} \mathbf{W}^2_{jl}}{\boldsymbol{SNR}^*[j]}$
4. Generate the latent factors $\boldsymbol{\gamma}_i \in \mathbb{R}^r$ associated with $\mathbf{X}^{(1)}_{i\cdot}$: $\boldsymbol{\gamma}_i \overset{\text{iid}}{\sim} N(\mathbf{0}, \mathbf{I}_r), i = 1, 2, \cdots, n$. We used $r = 3$ in experiments.
5. Generate the relevant features as $(\mathbf{X}^{(1)}_{i\cdot})^T = \mathbf{W}\boldsymbol{\gamma}_i + \mathbf{e}^{(1)}_i,\ i = 1, 2, \cdots, n$.

For the noisy variables, we assume the signal is equal to 0. Therefore, the generation of noise is sufficient. Following are the steps to generate $\mathbf{X}^{(2)}_{n\times d_{noise}}$:

1. Generate noise variances, $\boldsymbol{\Psi}^{*(2)} = \text{diag}(\sigma^{*2}_{11}, \sigma^{*2}_{12}, \cdots, \sigma^{*2}_{d})$ and
$$\sigma^{*2}_{(10+j)} \overset{\text{iid}}{\sim} Uniform(r/1.4, r/0.5),\ ;\ j = 1, \cdots, d_{noise}$$
2. Generate $(\mathbf{X}^{(2)}_{i\cdot})^T \overset{\text{iid}}{\sim} N(0, \boldsymbol{\Psi}^{*(2)}), i = 1, \cdots, n$.

Also, let us denote $\boldsymbol{\Psi}^* = [\boldsymbol{\Psi}^{*(1)}, \boldsymbol{\Psi}^{*(2)}]$ and $\boldsymbol{\psi}^* =$ diagonal elements of $(\boldsymbol{\Psi}^*)$ and
$\boldsymbol{sig}^* =$ diagonal elements of $(\mathbf{W}\mathbf{W}^T)$ for future use.

Smaller SNRs usually correspond to larger error variances. Therefore, the true SNRs range from 0.5 (small) to 1.4 (large) to make true feature recovery more challenging. The noise variable variances for the irrelevant dimensions are made comparable to those of the signal dimensions using the uniform distribution, as specified above.



(a) (b)

Figure 2.1: Generated plots using $n = 1000$ and $d = 110$, for $err(\hat{\boldsymbol{sig}})$ in (a), $err(\hat{\boldsymbol{\psi}})$ in (b)

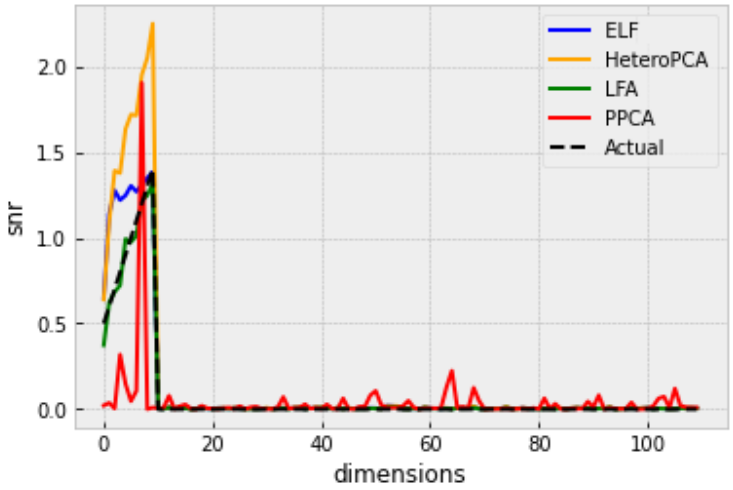


Figure 2.2: Comparison of $\boldsymbol{S\hat{N}R}$ vs $\boldsymbol{S\hat{N}R}^*$ for $n = 1000$ and $d = 110$

Parameter Estimation Evaluation. We compare the estimation error of the model parameters for the four SNR-based methods (i.e., PPCA, LFA, ELF, HeteroPCA).

Within a simulated dataset, we analyzed the estimation error for the signals ($\hat{\boldsymbol{sig}}$) and error variances ($\hat{\boldsymbol{\psi}}$) across multiple dimensions (denoted as d) for various iteration counts. For an estimate $\hat{\boldsymbol{\theta}}_{d\times 1}$ corresponding to the parameter, $\boldsymbol{\theta}^*_{d\times 1}$, we measure the mean absolute deviation (MAD) between those two over the d dimensions, denoted as $err(\hat{\boldsymbol{\theta}})$. It is defined below:

$$err(\hat{\boldsymbol{\theta}}) = MAD(\hat{\boldsymbol{\theta}}) = \frac{1}{d}\sum_{i=1}^{d} |\boldsymbol{\theta}^*_i - \hat{\boldsymbol{\theta}}_i|, \tag{2.129}$$

where $\hat{\boldsymbol{\theta}}$ could be one of $(\boldsymbol{sig}, \boldsymbol{\psi})$.

Figure 2.1 displays the estimation errors of the signal and noise variances as $err(\hat{\boldsymbol{sig}})$ and $err(\hat{\boldsymbol{\psi}})$ respectively for different iteration counts for all the SNR based methods. In contrast to other methods, which stabilize at a value higher than 0, those of the LFA method consistently decrease, eventually converging towards zero within 100 iterations. The performance of ELF and HeteroPCA methods closely resembles each other. ELF exhibited slightly superior performance over HeteroPCA in estimating the signal ($\boldsymbol{sig}$), while HeteroPCA outperformed ELF in estimating $\boldsymbol{\psi}$. Both methods reached stability within the initial 5 iterations. The PPCA method exhibits the maximum estimation error for $\hat{\boldsymbol{sig}}$ among all the SNR-based methods. However, the estimation of $\hat{\boldsymbol{\psi}}$ is considerably lower than the ELF and HeteroPCA methods.

In Figure 2.2 are plotted the estimated SNRs $\boldsymbol{\hat{SNR}}$ along with the true SNRs $\boldsymbol{SNR}^*$ for all SNR-based methods. All the methods provided estimates close to 0 for the noisy dimensions (i.e., $11, 12, \cdots, 110$) except for PPCA. There are several dimensions with significant noise levels where PPCA provided $\boldsymbol{\hat{SNR}}$ considerably above 0, whereas there are several crucial dimensions where PPCA produced $\boldsymbol{\hat{SNR}}$ near zero. From the plot, it's evident that LFA has the most accurate estimation of $\boldsymbol{SNR}^*$. Regarding ELF, we observed that its $\boldsymbol{\hat{SNR}}$ values are above 0 and close to corresponding positive values of $\boldsymbol{SNR}^*$, although they do not precisely align with the values of $\boldsymbol{SNR}^*$ as effectively as LFA does. Conversely, HeteroPCA tends to overestimate the positive values of $\boldsymbol{SNR}^*$ the most, compared to other methods. However, $\boldsymbol{\hat{SNR}}$ provided by HeteroPCA still captures the pattern of $\boldsymbol{SNR}^*$ more efficiently than ELF.

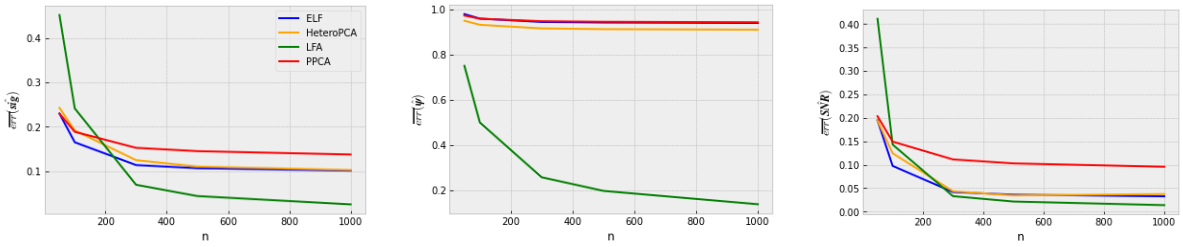
Furthermore, we have simulated 50 datasets for various sample size choices (denoted as n), to measure the estimation errors for ($\hat{\boldsymbol{sig}}$,$\hat{\boldsymbol{\psi}}$,$\boldsymbol{\hat{SNR}}$) using the average of MAD over d dimensions. We have already defined $err(\hat{\boldsymbol{\theta}})$ in (2.129) as the mean AD over d-dimensions. For multiple datasets with fixed value of n and $d = 110$, we will use the average of $err(\hat{\boldsymbol{\theta}})$,denoted as $\overline{err}(\hat{\boldsymbol{\theta}})$. Let $err_i(\hat{\boldsymbol{\theta}}), i = 1, 2, \cdots p$ be the estimation errors using p different samples. Then $\overline{err}(\hat{\boldsymbol{\theta}})$ is defined as:

$$\overline{err}(\hat{\boldsymbol{\theta}}) = \frac{1}{p}\sum_{i=1}^{p} err_i(\hat{\boldsymbol{\theta}}). \tag{2.130}$$

In our case, $p = 50$.

Figure 2.3 presents the average estimation error $\overline{err}(\hat{\theta})$ plotted against the sample size n, where θ belongs to the set of parameters $\{\boldsymbol{sig}, \boldsymbol{\psi}, \boldsymbol{SNR}\}$.

Across all three graphs, an observable trend is evident, suggesting that the LFA method provides maximum reliability as the sample size increases. Specifically, for LFA, as we gather more data,



(a) (b) (c)

Figure 2.3: Generated plots using $d = 110$ and different values of n, for $\overline{err}(\hat{\boldsymbol{sig}})$ in (a), $\overline{err}(\hat{\boldsymbol{\psi}})$ in (b) and $\overline{err}(\hat{\boldsymbol{SNR}})$ in (c)

the errors in estimating all parameters decrease. Notably, the LFA method exhibits the smallest estimation error across all parameters when compared to the other methods.

Additionally, the ELF and HeteroPCA methods, both nonparametric variants of LFA, operate on SNR. The performance of these two methods is very close in terms of estimation error corresponding to the set of parameters $\{\boldsymbol{sig}, \boldsymbol{\psi}, \boldsymbol{SNR}\}$. Conversely, PPCA represents a parametric alternative, predicated upon the assumption of uniform error variance across dimensions. In Figure 2.3 (a), the ELF signal estimates exhibit the smallest approximation error after LFA, and is marginally better than HeteroPCA. The PPCA provides the largest average error for estimating $\hat{\boldsymbol{sig}}$ among all other methods.

In Figure 2.3 (b), HeteroPCA shows a slight advantage by consistently providing smaller $\overline{err}(\hat{\boldsymbol{\psi}})$ values across all sample sizes compared to ELF and PPCA, with the latter two exhibiting similar performance. LFA stands out as significantly superior in estimating $\boldsymbol{\Psi}$ compared to all other methods.

In Figure 2.3 (c), LFA, HeteroPCA, and ELF demonstrate comparable performance, with ELF and HeteroPCA showing slightly higher $\overline{err}(\hat{\boldsymbol{SNR}})$ values than LFA. Conversely, PPCA falls behind due to its assumption of homoscedasticity, which poses a challenge in accurately estimating model parameters.

Table 2.1 displays the average (standard deviation) of $err_i(\hat{\theta}), i = 1, 2, \cdots, 50$ for varying sample sizes n. Overall, the mean values for PPCA tend to be higher compared to those of the other methods. Additionally, we observed that the standard deviations of $err_i(\hat{\boldsymbol{SNR}}), i = 1, 2, \cdots, 50$, for small values of n using the LFA method are notably higher than those of other methods, which eventually decrease as the sample size increases.

Figures 2.4 and 2.5 provide a comprehensive empirical validation of our theoretical analysis. Figure 2.4 showcases the comparison between $\overline{err}(\hat{\boldsymbol{sig}})$ and the corresponding theoretical bound

Table 2.1: Mean and standard deviation of parameter estimate errors for SNR based methods.

		Mean (std)			
	n	ELF	HeteroPCA	LFA	PPCA
$\hat{sig}$	50	**0.23**(0.022)	0.24(0.014)	0.45(0.568)	**0.23**(0.007)
	100	**0.17**(0.019)	0.19(0.015)	0.24(0.217)	0.19(0.006)
	300	0.11(0.003)	0.12(0.008)	**0.07**(0.007)	0.15(0.003)
	500	0.11(0.002)	0.11(0.003)	**0.04**(0.004)	0.15(0.002)
	1000	0.10(0.002)	0.10(0.004)	**0.03**(0.003)	0.14(0.001)
$\hat{\Psi}$	50	0.98(0.014)	0.95(0.01)	**0.74**(0.038)	0.97(0.01)
	100	0.96(0.014)	0.93(0.009)	**0.50**(0.044)	0.96(0.007)
	300	0.94(0.002)	0.91(0.007)	**0.26**(0.008)	0.95(0.003)
	500	0.94(0.001)	0.91(0.002)	**0.20**(0.006)	0.94(0.002)
	1000	0.94(0.001)	0.91(0.001)	**0.14**(0.005)	0.94(0.002)
$\hat{SNR}$	50	**0.19**(0.002)	**0.19**(0.002)	0.41(0.059)	0.20(0.002)
	100	**0.09**(0.002)	0.12(0.001)	0.14(0.04)	0.15(0.001)
	300	0.04(0.001)	0.04(0.001)	**0.03**(0.021)	0.11(0.001)
	500	0.04(0.001)	0.04(0.001)	**0.02**(0.015)	0.11(0.001)
	1000	0.03(0.001)	0.03(0.001)	**0.01**(0.009)	0.10(0.001)

vs n on a log-log scale for PPCA. This figure empirically validates our non-asymptotic analysis for the signal variance estimator, $\boldsymbol{sig}$. The near-linear slope of both curves on the log-log scale visually demonstrates the expected $1/\sqrt{n}$ rate of convergence. Similarly, Figure 2.5 illustrates the average estimation error, its corresponding theoretical bound, and the systematic bias for the LFA estimators ($\boldsymbol{sig}, \boldsymbol{\psi}, \boldsymbol{SNR}$) vs. the n. Across all three subplots, the key observations confirm the soundness of our framework: the empirical error consistently lies below our derived theoretical bound, demonstrating that our bound is valid and correctly upper-bounds the true error. A closer comparison reveals important structural details. The estimation behaviors for the ($\boldsymbol{sig}, \boldsymbol{\psi}$) are nearly identical, as shown by the parallel trends in plots (a) and (b). Furthermore, the plots highlight the significance of the systematic bias (green dashed line). This bias constitutes a substantial component of the overall error. It converges much more slowly than the stochastic error, underscoring the need to analyze it as a distinct, non-vanishing term. While the convergence rate of the empirical $\boldsymbol{SNR}$ error is comparable to that of the signal and noise, its structure is different. The lower magnitude of its bias term relative to its overall bound suggests that the final error is more heavily influenced by the stochastic component, which diminishes rapidly with n.

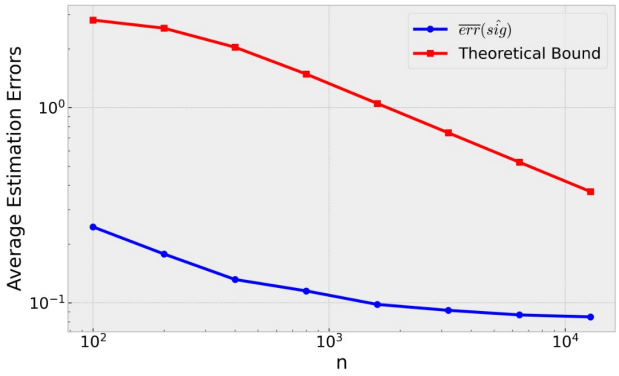
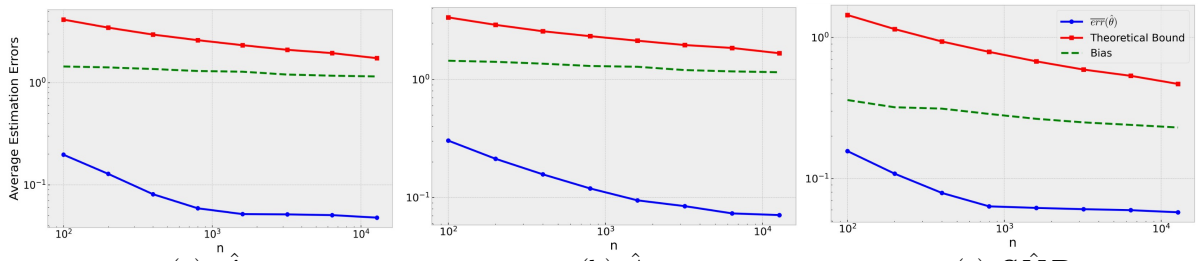


Figure 2.4: Estimation error and theoretical bound vs. number of observations (n) for the PPCA signal variance.



(a) $\hat{\boldsymbol{sig}}$ (b) $\hat{\boldsymbol{\psi}}$ (c) $\hat{\boldsymbol{SNR}}$

Figure 2.5: Estimation errors, Theoretical Bounds and Biases vs. number of observations (n) of **LFA** for (a) the signal variance, $\hat{\boldsymbol{sig}}$, (b) the noise variance, $\hat{\boldsymbol{\psi}}$, and (c) the SNRs, $\hat{\boldsymbol{SNR}}$, when $d = 110$.

Feature Recovery Evaluation. **Feature Recovery Evaluation.** In this experiment, we consider feature selection accuracy a pivotal aspect of method assessment. To measure how accurate the feature selection process is, we compare the set of indices of the features that are truly relevant(signals), denoted as $\mathcal{I}_{true}$ with the ones each method has predicted as relevant, denoted as $\mathcal{I}_{pred}$. The feature selection accuracy is defined as the average percentage of features that are correctly recovered:

$$Acc = E(|\mathcal{I}_{true} \cap \mathcal{I}_{pred}|)/|\mathcal{I}_{true}|, \tag{2.131}$$

where the expected value is computed over 50 independent runs. We conducted 50 simulations for each n and d_{noise} combination and recorded the Acc values in Table 2.2.

Our experimentation involved varying d_{noise} within $\{10, 50, 100\}$ and n within $\{50, 100, 300, 500, 1000\}$. Table 2.2 presents the accuracies of variable selection for six methods discussed in previous chapters. A clear and consistent trend across all methods is a direct relationship between sample size and

Table 2.2: Feature selection accuracy for outlier-free data.

Noise	n	Methods			
		PPCA	**LFA**	**ELF**	**HeteroPCA**
10	50	71.2	**90.6**	87.6	84.4
	100	82.4	**97.0**	94.0	93.0
	300	88.0	**100.0**	98.0	98.8
	500	92.4	**100.0**	99.0	99.2
	1000	95.6	**100.0**	99.2	99.8
50	50	55.4	70.4	**73.4**	65.6
	100	72.6	91.8	**92.8**	87.0
	300	79.8	**100.0**	98.6	94.4
	500	86.4	**100.0**	99.4	98.6
	1000	91.4	**100.0**	99.0	99.2
100	50	49.0	57.4	**59.0**	55.8
	100	62.8	**87.4**	87.2	75.6
	300	82.0	**99.6**	**99.6**	96.4
	500	82.6	**100.0**	99.4	95.2
	1000	90.0	**100.0**	99.4	99.6

accuracy. As the number of observations n increases from 50 to 1000, the feature recovery accuracy of all models improves significantly. This validates the statistical consistency of the SNR-based approach, confirming that with more data, the models become progressively better at distinguishing true signal from noise. Conversely, for a fixed sample size, increasing the number of noise features from 10 to 100 generally degrades performance, highlighting the challenge of identifying relevant signals in a higher-dimensional, noisier space.

The central finding of this simulation is the clear stratification in performance among the different generative models, with LFA demonstrating overwhelmingly superior performance. Across nearly all conditions, LFA achieves the highest accuracy, often reaching perfect (100%) or near-perfect feature recovery with only $n = 300$ samples, even in the most challenging scenario with 100 noise features. The ELF method proves to be a very strong second, with performance that is highly competitive with LFA, particularly in the low-sample-size regime ($n = 50$ and $n = 100$), where it occasionally outperforms all other methods. HeteroPCA also performs robustly, consistently surpassing PPCA, but it generally lags behind the top-tier performance of LFA and ELF.

CHAPTER 3

FEATURE SELECTION USING SPARSITY INDUCING PENALTIES

3.1 Related Work

The use of sparsity inducing penalties in high-dimensional data analysis is one of the most popular research directions. To select features in a high-dimensional dataset, sparse penalty-based methods reduce prediction errors by setting many feature coefficients to zero. This helps simplify the model.

In the case of binary classification or regression, one way of performing feature selection is to use a special penalty term called "p-norm sparsity-inducing penalty" on the coefficient matrix $\mathbf{W}$, where p can be any number from 0 to 1. The goal is to minimize a loss $L(\mathbf{W}) = loss(\mathbf{y}, \mathbf{WX}) + \alpha\|\mathbf{W}\|_p$. The penalty term encourages most features to be small or even zero. The parameter α helps balance between making accurate predictions versus simplifying the model. Even though $p = 0$ would be ideal in this case, it is not feasible for optimization. Often $p = 1$ is used instead, which usually results in a convex optimization problem. This method, called LASSO [109], has become popular among feature selection methods.

Sparsity has also been introduced into PCA methods, such as Sparse PCA [130]. Regular PCA identifies the most informative directions in the data, but these directions can involve many features while sparse PCA does the same by introducing sparsity. Sparse principal components often rely on only a few features, making them easier to interpret and potentially reducing model complexity. It also outperforms traditional PCA in the presence of data correlation. In recent years, sparse PCA has been widely used for feature selection across many fields [57, 21]. These algorithms seek sparse loading vectors separately and progress sequentially. The loading matrix obtained may lack optimality and contain too many variables. On the other hand, [102] proposes joint sparsity across all loading vectors, to ensure dimension reduction even when constructing a number of factors. This turns out to be particularly helpful in rank-constrained variable screening. We will discuss this approach in detail in the following section. In recent years, [78] has introduced sparse PCA by adding "False Discovery Rate" (FDR) control. FDR limits the chance of accidentally picking

irrelevant variables (false positives) while selecting the important ones. They use a tool called the T-Rex selector to achieve this, which automatically handles selection without requiring manual tweaking of how "sparse" (few variables) the model should be. [124] reformulates convex SPCA with PSD cone constraints for faster optimization via two-step PSD projection. They also include regularization (penalty) strategies to fine-tune sparsity.

Sparsity constraints have also been combined with other models to identify important features during a particular task. [49] tackle cancer classification using gene expression by introducing a hybrid $L_{\frac{1}{2}} + L_2$ regularization for sparse logistic regression. This technique leverages the L1 penalty for feature selection (finding key genes) and the L2 penalty for stability (grouping correlated genes). [12] proposes an online feature selection method suitable for massive datasets. It uses sparse gradients to promote sparsity in its feature weights during classifier training. Therefore, features with minimal influence will have their weights driven towards zero and will be removed from the model eventually.

3.2 Selective Reduced Rank Regression

In this section, we will introduce the Selective Reduced Rank Regression (SRRR) proposed by [102]. We will then propose a robust version of it in the following section. SRRR is an approach for extracting selective factors from a parsimonious set of features in a multivariate regression setup. First, we will describe the original problem of SRRR. Eventually, we will make some adjustments and add more constraints to make it suitable for feature selection in the unsupervised setup.

The original optimization problem (in a supervised setup) uses a low-rank representation of the data $\mathbf{X}_{n\times d}$ to predict a response matrix $\mathbf{Y}_{n\times p}$.

$$\min_{\boldsymbol{W}\in\mathbb{R}^{p\times d}} F(\mathbf{W},\lambda), \text{ where } F(\mathbf{W},\lambda) = \frac{1}{K}\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2 + \sum_{j=1}^{d} P(\|\mathbf{W}_j\|_2,\lambda), \text{ and } r(\mathbf{W}) \le r. \quad (3.1)$$

Here $\mathbf{W}$ is the loading matrix, λ and r are the regularization parameters that control the sparsity and rank of the loading matrix respectively, $r \ll \min\{p,d\}$, K denotes the scaling parameter for (3.1) and P is a sparsity promoting, possibly non-convex penalty function, which is associated with some thresholding function Θ defined in Definition 3 below. First we will introduce the definition of a thresholding function Θ.

Definition 2. *A **Thresholding Function** is a function $\Theta(\cdot;\lambda) : \mathbb{R} \to \mathbb{R}$ with $0 \le \lambda < \infty$ that satisfies:*

- $\Theta(-t;\lambda) = -\Theta(t;\lambda)$,
- $\Theta(t;\lambda) \leq \Theta(t';\lambda)$ *for* $t \leq t'$,
- $\lim_{t\to\infty} \Theta(t;\lambda) = \infty$,
- $0 \leq \Theta(t;\lambda) \leq t$ *for* $0 \leq t < \infty$.

$\Theta(t;\lambda)$ is an odd monotone unbounded shrinkage rule for t. A vector version, $\Theta(\mathbf{t};\lambda)$ is defined componentwise.

Definition 3. Θ ***induced P:*** *Given any thresholding function* $\Theta(\cdot;\lambda)$*, we say penalty* P *is induced by* Θ *if:*

$$\begin{aligned} P(t;\lambda) - P(0;\lambda) &= P_\Theta(t;\lambda) + q(t;\lambda), \\ P_\Theta(t;\lambda) &= \int_0^{|t|} [\Theta^{-1}(u;\lambda) - u]du, \\ \Theta^{-1}(u;\lambda) &= \sup\{s : \Theta(s;\lambda) \leq u\}, \end{aligned}$$

for some non negative $q(\theta,\lambda) : q\{\Theta(t;\lambda)\} = 0, t \in \mathbb{R}$.

The $\Theta-$ induced property enables us to substitute the penalty function P with the corresponding thresholding function Θ, to directly control the sparsity of each row in $\mathbf{W}$. This not only streamlines the feature selection process but also enhances the model's interpretability. This optimization problem is a modified version of reduced rank regression[54], which aims to find a low rank solution for $\mathbf{W}$, while considering two parsimonies jointly: 1) low rank constraint of $\mathbf{W}$ and 2) sparsity constraint on $\mathbf{W}$.

The author shows that the proposed method enjoys sharp oracle inequalities even when the number of input features is much larger than the number of response variables.

Theorem 11. *[due to [102]] Let* $\boldsymbol{Y_{n\times p} = X_{n\times d} W^* + E}$*, with all entries of* $\boldsymbol{E}$*, independent and identically distributed as* $\mathcal{N}(0,\sigma^2)$*. Let* $\hat{\boldsymbol{W}}$ *be a selective reduced rank regression estimator that minimizes equation* $\|\mathbf{Y} - \mathbf{XW}\|_F^2 + \lambda^2\|\mathbf{W}\|_{2,0}$*, subject to* $r(\mathbf{W}) \leq r$*. Then, under* $\lambda = A\sigma(r + \log(d))^{\frac{1}{2}}$*, where A is a large enough constant, the following oracle inequality holds for any* $\boldsymbol{W} \in \mathbb{R}^{d\times p}$ *with* $r(\boldsymbol{W}) \leq r$*:*

$$E(\|\boldsymbol{X\hat{W} - XW^*}\|_F^2) \lesssim \|\boldsymbol{XW - XW^*}\|_F^2 + \lambda^2\|\mathbf{W}\|_{2,0} + (p-r)r\sigma^2 + \sigma^2. \tag{3.2}$$

Here, "$\lesssim$*" means that the inequality holds up to a multiplicative constant.*

Theorem 11 establishes non-asymptotic oracle inequalities for the prediction error of the selective reduced rank regression estimator. The estimator minimizes the squared Frobenius norm loss $\|\mathbf{Y} - X\mathbf{W}\|_F^2$ augmented with group sparsity penalties on the coefficient matrix $\mathbf{W}$, subject to a rank constraint $r(\mathbf{W}) \leq r$. $\mathbb{E}(\|X\hat{\mathbf{W}} - \mathbf{X}\mathbf{W}^*\|_F^2)$ is bounded, up to a universal constant, by $\|\mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}^*\|_F^2$ (bias) plus a penalty term involving the number of non-null rows, plus $(p-r)r\sigma^2 + \sigma^2$. The resulting error rate for the true model ($\mathbf{W} = \mathbf{W}^*$) is $O((\|\mathbf{W}^*\|_{2,0} + p - r^*)r^* + \|\mathbf{W}^*\|_{2,0} \log p)$, which is sharper than rates computed in several other contemporary works [19, 77].

This theorem is crucial because it rigorously justifies the benefit of *joint* variable selection and rank reduction, demonstrating that simultaneous regularization achieves lower prediction error than applying either technique alone.

This joint regularization simultaneously controls the number of active predictors via λ and the dimensionality of the factor space via r, enabling interpretable factor extraction from high-dimensional multivariate data. To adaptively tune these parameters without cross-validation, the author introduces the predictive information criterion (PIC), defined as $P_o(\mathbf{W}) = \sigma^2\big[\{q \wedge r(\mathbf{W}) + p - r(\mathbf{W})\}r(\mathbf{W}) + J(\mathbf{W})\log(ed/J(\mathbf{W}))\big]$, where $q = r(\mathbf{X})$,$J(\mathbf{W}) = \|\mathbf{W}\|_{2,0}$ and σ^2 is the noise variance. The PIC integrates a degrees-of-freedom term for rank reduction with a risk inflation term for variable selection uncertainty. The paper established a non-asymptotic oracle inequality showing that minimizing $\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 + AP_o(\mathbf{W})$ yields prediction error within a constant factor of the minimax optimal rate over all candidate models, without assumptions on $\mathbf{X}$ or $\mathbf{W}^*$. A scale-free variant of $P_o(\mathbf{W})$ eliminates σ^2 estimation, ensuring practical applicability. Unlike BIC or cross-validation, which lack theoretical support in joint sparse low-rank settings, the PIC is minimax optimal and naturally adapts to unknown sparsity and rank structures. This scale free version of PIC is given by: $\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2/\{pn - AP_o(\mathbf{W})/\sigma^2\}$

The $\Theta-$induced penalties enable a universal algorithmic treatment via iterative thresholding [101], with convergence guarantees even for nonconvex penalties. Examples include: (i) *convex* penalties such as the group $l_1(\Theta(s;\lambda) = (s-\lambda)_+)$, which induces $(P(s;\lambda) = \lambda s)$; (ii) *nonconvex* penalties like SCAD[31], MCP[123] etc., all of which induce hard-thresholding-like behavior$(\Theta(s;\lambda) = s \cdot \mathbf{I}(|s| \geq \lambda))$ for large signals while smoothly shrinking small ones. Algorithm 4 demonstrates the iterative optimization process for the SRRR problem.

Algorithm 4: Selective Reduced Rank Regression Methods (Supervised Case):

Input :

- Rank r , $1 \leq r \leq p$ and thresholding parameter $\lambda : \lambda \geq 0$.
- Θ : Thresholding rule
- M_{inner} : Maximum number of inner iterations
- M_{outer} : Maximum number of outer iterations

Output : Estimated matrices $\hat{\mathbf{W}} = \mathbf{W}_{(t)}$, $\hat{\boldsymbol{S}} = \boldsymbol{S}_{(t)}$,$\hat{\boldsymbol{V}} = \boldsymbol{V}_{(t)}$

Initialize: Reduced Rank Regression Estimate has been used here to initialize.

- $\mathbf{V}_{(0)} = \mathbf{V}_r, \mathbf{V}_r$ is formed by first r eigen vectors of $\mathbf{Y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^+\mathbf{X}^T\mathbf{Y}$
- $\mathbf{S}_{(0)} = (\mathbf{X}^T\mathbf{X})^+\mathbf{X}^T\mathbf{Y}\mathbf{V}_r$
- $\mathbf{W}_{(0)} = (\mathbf{X}^T\mathbf{X})^+\mathbf{X}^T\mathbf{Y}\mathbf{V}_r\mathbf{V}_r^T$

Calculate $K = \|\boldsymbol{X}\|_2^2$
for $t = 0$ *to* M_{outer} **do**
    Calculate $\boldsymbol{M} = \boldsymbol{Y^T X S_{(t-1)}}$, compute the reduced rank $\boldsymbol{U_r D_r V_r^T} = \boldsymbol{M}$ by SVD
    Compute $\boldsymbol{V_{(t)}} = \boldsymbol{U_r V_r^T}$
    To update $\boldsymbol{S}$, set $l = 0$, $\tilde{\boldsymbol{S}}_{(0)} = \boldsymbol{S}_{(t-1)}$
    for $l = 0$ *to* M_{inner} **do**
        Compute $\Xi_{(l,t)} = \boldsymbol{X^T Y V_{(t-1)}/K + (I - X^T X/K)\tilde{S}}_{(l-1)}$
        $\tilde{\boldsymbol{S}}_{(l)} = \Theta(\Xi_{(l,t)}, \lambda)$
        if $\{\|\tilde{\boldsymbol{S}}_{(l)} - \tilde{\boldsymbol{S}}_{(l-1)}\|$ is sufficiently small$\}$ break
    Compute $\hat{\boldsymbol{S}}_{(t)} = \tilde{\boldsymbol{S}}_{(l)}$
    Compute $\hat{\boldsymbol{W}}_{(t)} = \boldsymbol{S_{(t)} V_{(t)}^T}$
    if $\{\|\hat{\boldsymbol{W}}_{(t)} - \hat{\boldsymbol{W}}_{(t-1)}\|_2$ is sufficiently small$\}$ break

3.3 Feature Selection for Selective PCA

SRRR stands as a valuable tool for feature selection, efficiently optimizing the problem defined in Eq. (3.1), by integrating several parsimonies: low-rank constraint and row-wise sparsity. In our methodological framework, we do not include class label information in the optimization problem, thereby enhancing computational efficiency and scalability. In this unsupervised setup, the data matrix is $\mathbf{X} \in \mathbb{R}^{n\times d}$, and the design matrix can be assumed to be $\mathbf{I}$. The new method has been named 'Selective Principal Component Analysis'. The objective function along with variable screening is formulated in the following way, with sparsity control and variable screening constraints:

$$\min_{\mathbf{S}\in\mathbb{R}^{d\times r},\mathbf{V}\in\mathbb{O}^{n\times r}} \frac{1}{2K}\|\mathbf{X}-\mathbf{S}\mathbf{V}^\top\|_F^2+\frac{\eta}{2}\|\mathbf{S}\|_F^2 \quad \text{subject to} \quad \|\mathbf{S}\|_{2,0}\le m. \tag{3.3}$$

The matrix $\mathbf{V}$ can be regarded as the unobserved latent factor matrix, which is accountable for variation in the data matrix $\mathbf{X}$, and $\mathbf{S}$ can be interpreted as the coefficient matrix, which transforms the r-dimensional latent vector into a d-dimensional observation. Therefore, $\hat{\mathbf{S}}$ plays an important role in the variable selection procedure.

Here, $rank(\mathbf{S}) = r \ll m$ (number of selected features) facilitates the projection of the chosen features into a lower-dimensional space, as the chosen features may not be independent from each other. Additionally, the low-rank constraint ensures a reduction in the effective number of parameters, thereby improving estimation efficiency.

Furthermore, the cardinality constraint on $\mathbf{S}$, rather than a penalty, enables the direct control of the number of predictors selected and is very intuitive. One can use the quantile thresholding function θ to optimize the new objective function to handle the row-wise sparsity.

Quantile Thresholding: Given $1 \le i \le d$, for any $\boldsymbol{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots \mathbf{s}_d)^T \in \mathbb{R}^{d\times r}$, to get m features, the thresholding function can be defined as,

$$\Theta(\boldsymbol{s},\eta,m) = \begin{cases} \boldsymbol{s}_{(j)}/(1+\eta) & \text{if } 1\le j\le m,\\ 0 & \text{otherwise.}\end{cases} \tag{3.4}$$

Here $\{\boldsymbol{s}_{(i)}, i = 1, 2, \cdots, d\}$ are the ordered row vectors of $\mathbf{S}$ based on $\|\boldsymbol{s}_i\|_2$. To get m features, we will use $\Theta(\Xi_{(l,t)}, m)$ in Algorithm 5.

We will exclude a feature from $\mathbf{X}$ if it corresponds to an entire row of $\hat{S}$ set to 0, determined by the thresholding rule. The computation becomes much lighter in the unsupervised case for the identity design matrix. Note that the Selective PCA enjoys all the theoretical properties of SRRR,

as the oracle theorem or any other theorem following that in [102], does not impose any condition on $\mathbf{X}, \mathbf{W}^*$.

Algorithm 5: Feature Selection with Selective PCA

Input : Rank r, $1 \leq r \leq d$, desired number of features m, maximum number of iterations M_{iter}.

Output : Set $\mathcal{I}$ of indices corresponding to the m most important features

Initialize: Initialize $\mathbf{S}, \mathbf{V}$ and $\mathbf{W}$ similar to Algorithm 4 with $\mathbf{Y} = \mathbf{X}^T$ and $\mathbf{X} = \mathbf{I}$

for $t = 1$ *to* M_{iter} **do**

Calculate $\boldsymbol{M = X S_{(t-1)}}$, compute the reduced rank SVD, $\boldsymbol{M = U_r D_r V_r^T}$

Compute $\boldsymbol{V_{(t)} = U_r V_r^T}$

Compute $\boldsymbol{\Xi_{(t)} = X^T V_{(t-1)}}$

Calculate $\mathbf{S}^{(t)} = \Theta(\boldsymbol{\Xi_{(t)}}; m)$ using Eq. (3.4).

Compute $\hat{\boldsymbol{W}}_{(t)} = \boldsymbol{S_{(t)} V_{(t)}^T}$

if $\{\|\hat{\boldsymbol{W}}_{(t)} - \hat{\boldsymbol{W}}_{(t-1)}\|_2$ is sufficiently small$\}$ break

Obtain selected set of indices corresponding to the m most important features, $\mathcal{I}_m = \{i : \mathbf{s}_i \neq \mathbf{0}\}$

3.4 Robust Loss Minimization

We will now propose a robust version of Selective PCA to handle extreme values in high-dimensional datasets effectively, along with feature selection and rank optimization. The mean squared error (MSE) is sensitive to outliers as the squared differences can be heavily influenced, leading to significant errors and affecting the model's overall performance. Robust loss functions, on the other hand, are designed to handle outliers more effectively by being less sensitive to extreme values. Here are some challenges with the MSE loss in the presence of outliers and some ways in which robust loss functions can address these issues:

- The MSE loss amplifies the influence of outliers by squaring errors, but robust loss functions mitigate this effect by downweighting the impact of significant errors.
- The optimization process with the MSE loss becomes unstable due to the linear increase in gradient with outliers. In contrast, robust loss functions can limit such influence on the gradient to prevent extensive updates.

To obtain robust estimates of location, the M-estimation method in linear regression setup was proposed by [50]. Instead of minimizing $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$, in a linear regression setup, the author

suggested using a general loss function ρ, and a robust location estimate can be found by minimizing the following:

$$\min_{\boldsymbol{\beta}} \sum_i \rho(\mathbf{y}[i] - \mathbf{x}_i^T \boldsymbol{\beta}). \tag{3.5}$$

Here, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)^T$. As a consequence, we replace the following score equation in OLS:

$$\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) = 0$$

with the corresponding score equation in the M-estimation method:

$$\sum_i \mathbf{x}_i \psi(\mathbf{y}[i] - \mathbf{x}_i^T \boldsymbol{\beta}) = 0,$$

where $\psi(x) = \frac{\partial}{\partial x}\rho(x)$. But the score function ψ can be defined more generally than a derivative function.

Here, we are going to use a Lorentzian loss [17] as ρ and explore feature selection methods similar to the Selective PCA setup afterwards. Originally, the Lorentzian loss was defined as:

Definition 4 (due to [17])**.** *The **Lorenzian Loss** is defined as:*

$$\rho_L(r, \sigma^2) = \log(1 + \frac{r^2}{\sigma^2}).$$

The Lorenzian loss function with $\sigma^2 = 2c$, where $c > 0$ is a scaling factor, can be viewed as a special case of the following general family of robust loss functions [14].

Definition 5. *[due to [14]] The general family of robust loss function $\rho(x, \alpha, c)$ is defined as:*

$$\rho(x, \alpha, c) = \begin{cases} \frac{1}{2}(\frac{x}{c})^2 & \text{if } \alpha = 2, \\ \log(\frac{1}{2}(\frac{x}{c})^2 + 1) & \text{if } \alpha = 0, \\ 1 - \exp(-\frac{1}{2}(\frac{x}{c})^2) & \text{if } \alpha = -\infty, \\ \frac{|\alpha-2|}{\alpha}((\frac{(\frac{x}{c})^2}{|\alpha-2|} + 1)^{\frac{\alpha}{2}} - 1) & \text{otherwise.} \end{cases}$$

Therefore, the corresponding gradients $\psi(x, \alpha, c)$ are:

$$\frac{\partial}{\partial x}\rho(x, \alpha, c) = \psi(x, \alpha, c) = \begin{cases} \frac{x}{c^2} & \text{if } \alpha = 2, \\ \frac{2x}{x^2+2c^2} & \text{if } \alpha = 0, \\ \frac{x}{c^2}\exp(-\frac{1}{2}(\frac{x}{c})^2) & \text{if } \alpha = -\infty, \\ \frac{x}{c^2}(\frac{(\frac{x}{c})^2}{|\alpha-2|} + 1)^{\frac{\alpha}{2}-1} & \text{otherwise.} \end{cases}$$

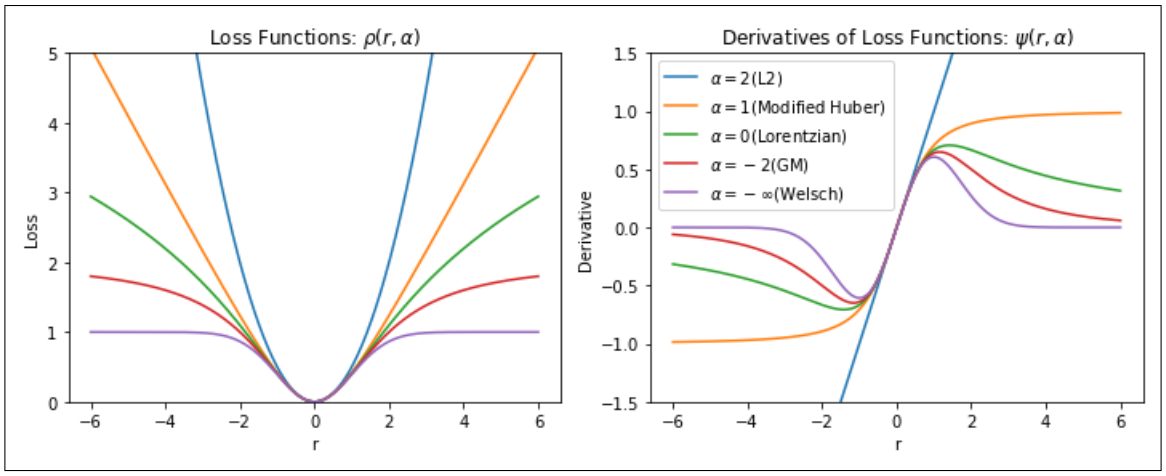


Figure 3.1: Several loss functions (left) and their corresponding derivatives (right) from Definition 5.

This family of loss functions is smooth with respect to x and has bounded first and second order derivatives for $\alpha \leq 1$. More specifically,

$$|\frac{\partial \rho}{\partial x}(x, \alpha, c)| \leq \begin{cases} \frac{1}{c}\left(\frac{\alpha-2}{\alpha-1}\right)^{\left(\frac{\alpha-1}{2}\right)} \leq \frac{1}{c} & \text{if } \alpha \leq 1, \\ \frac{|x|}{c^2} & \text{if } \alpha \leq 2. \end{cases}$$

To get an idea about how the robust losses behave, it is crucial to look at the corresponding gradients ψ.

- $\alpha = 2$ (L2 loss): In this case, the gradient ψ is linear. It means a larger error yields a larger gradient, which is not a desirable property of a robust loss function.
- $\alpha = 1$ (Charbonnier loss [25], pseudo-Huber loss [52]): The ψ function saturates, which means the large errors have as much effect as moderate errors on the gradients.
- $\alpha = 0$ (Lorentzian Loss [17]): The ψ begins to redescend toward 0 as the error gets larger. This means the large errors have less influence than the moderate errors.
- $\alpha < 0$ (Geman-McClure loss [34], Welsch loss [27, 68]): As α gets smaller, the rate at which ψ redescends towards 0 increases. Therefore, the effects of outliers on the gradient become even smaller.

Therefore, it is evident that the Lorenzian loss is more robust to outliers compared to many other loss functions with $\alpha > 0$. Therefore, if the data contains a significant number of outliers, the Lorenzian loss might be a better choice to minimize their influence on the loss calculation as $\psi(t)/t$ decreases sharply.

Now, we will define finite sample breakdown points, which are a crucial measure for quantifying the robustness of an estimator in the presence of outliers in the dataset. It can be defined in many ways, but we will use the definition from [51].

Definition 6. ***Finite Sample Breakdown Points*** *[due to [51]] Let* $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$ *be a finite sample of size n. We can corrupt this sample by performing* ϵ *contamination: Let* $\mathbf{y} = \{y_1, y_2, \cdots, y_m\}$ *be m arbitrary values, then the corrupted sample is* $\mathbf{x}' = \mathbf{x} \cup \mathbf{y}$ *of size* $m + n$ *and contains a fraction* $\epsilon = \frac{m}{m+n}$ *of bad values. Let* $T(\mathbf{x})$ *be a robust estimator and* $b(\epsilon, \mathbf{x}, T)$ *be the maximum bias associated with it. Therefore,* $b(\epsilon, \mathbf{x}, T) = \sup \|T(\mathbf{x}') - T(\mathbf{x})\|$*, where the supremum is taken over all* ϵ*-corrupted samples* $\mathbf{x}'$*.*

The finite sample breakdown point ϵ^* *is defined as:*

$$\epsilon^*(\mathbf{x}, T) = \inf\{\epsilon | b(\epsilon, \mathbf{x}, T) = \infty\}$$

The breakdown point can take the highest value of 1 for a constant statistic or a Bayes estimate with a prior that has compact support. It can also approach 0, for example, when T is the sample mean.

- **Sample Mean:** The sample mean has a breakdown point of $\epsilon = 1/n$. Replacing just a single data point with an infinitely large value will cause the mean to become infinite. As $n \to \infty$, its breakdown point is 0. It is not robust.
- **Sample Median:** The sample median has a breakdown point of $\epsilon \approx 1/2$. To make the median arbitrarily large, one must corrupt at least half of the data points. This is the highest possible breakdown point for any translation-equivariant location estimator, making it highly robust.

The breakdown point of an M-estimator is directly related to its score function ψ.

- M-estimators with **monotone** ψ-functions (like Huber's) have a breakdown point that is positive but strictly less than $1/2$.
- M-estimators with **redescending** ψ-functions (like the Lorentzian loss) can achieve the optimal breakdown point of $1/2$. This is because their ability to ignore extreme outliers completely prevents those outliers from driving the estimate to infinity.

There are multiple estimation methods available in the literature to obtain estimates with high breakdown points. Some of them are the Least median of squares (LMS)[91], Least trimmed squares (LTS)[91], and Least trimmed absolute values (LTA)[44]. All these estimation methods provide

robust estimates of location with high breakdown points, but they also have some drawbacks. LMS is proven to be less efficient statistically compared to LTS. This means that a larger sample size is required to arrive at the same conclusion in probabilistic terms when the distribution of errors is normal, due to its low statistical efficiency relative to LTS.

Redescending $\boldsymbol{\psi}$ in the M estimation methods yields estimates with potentially high breakdown. Like the trimming estimators LMS, LTS, and LTA, they can ignore observations that appear to deviate from the model. One of the main advantage of using Redescending $\boldsymbol{\psi}$ is, unlike the trimming estimators, the data drive the amount of trimming; only cases with extreme residuals will be trimmed.

Theorem 12 (due to [51])**.** *Let ρ be a loss function satisfying the following properties:*

- *ρ is symmetric,*
- $\rho(0) = 0$*,*
- $\lim_{|x|\to\infty} \rho(x) = \infty$*,*
- $\lim_{|x|\to\infty} \frac{\rho(x)}{x} = 0$*.*

Further, we assume that the corresponding ψ is continuous and there exists an x_0 such that $\psi(x)$ is weakly increasing for $0 \le x \le x_0$ and weakly decreasing for $x_0 \le x \le \infty$. Then, the ϵ-contamination breakdown point of an M-estimate is $\frac{1}{2}$.

Therefore, the robust loss function and its corresponding derivative that we will be working with are the following:

$$\rho_L(x) = \log(1 + \frac{1}{2}x^2) \tag{3.6}$$

$$\psi_L(x) = \frac{2x}{2 + x^2} \tag{3.7}$$

When ρ_L is applied to a matrix $\mathbf{M}$, we will perform the following operation to calculate the output:

$$\rho_L(\mathbf{M}_{n\times d}) = \sum_{i=1}^{n}\sum_{j=1}^{d} \log(1 + \frac{1}{2}M_{ij}^2).$$

Our initial optimization goal is to find a low-rank r and row-wise sparse matrix $\mathbf{W}$ that minimizes the given loss. It can be defined in the following way:

$$\begin{aligned}
&\min_{\boldsymbol{W}\in\mathbb{R}^{n\times d}} F(\boldsymbol{W}) \text{ with } \|\mathbf{W}\|_{2,0} \le m, r(\mathbf{W}) \le r \\
&= \min_{\boldsymbol{W}\in\mathbb{R}^{n\times d}} \rho_L(\mathbf{X} - \mathbf{W}) + \frac{\lambda}{2}\|\mathbf{W}\|_F^2 \text{ with } \|\mathbf{W}\|_{2,0} \le m, r(\mathbf{W}) \le r \\
&= \min_{\boldsymbol{S}\in\mathbb{R}^{d\times r}, \boldsymbol{V}\in\mathcal{O}^{n\times r}} \rho_L(\mathbf{X} - \mathbf{V}\mathbf{S}^T) + \frac{\lambda}{2}\|\mathbf{S}\|_F^2 \text{ with } \|\mathbf{S}\|_{2,0} \le m.
\end{aligned} \tag{3.8}$$

Although the employed loss exhibits nice robustness properties, its nonconvex nature makes it difficult to optimize. We will implement a surrogate function to optimize the problem in Eq. (3.8).

Definition 7. ***Surrogate Function.*** *Given* $\operatorname{argmin}_\beta f(\beta)$*, a surrogate function is defined as* $g(\beta, \beta^-)$ *with the following properties:*

p1) $g(\beta, \beta^-) \geq f(\beta)$,

p2) $g(\beta^-, \beta^-) = f(\beta^-)$.

We can reformulate the original problem using a surrogate function, and the predefined properties (p1 and p2) ensure the convergence of the optimization algorithm. Let us define $\beta^{t+1} = \operatorname{argmin}_\beta g(\beta, \beta_t)$. It can be easily seen that:

$$f(\beta_{t+1}) \leq g(\beta^{t+1}, \beta_t) \leq g(\beta^t, \beta_t) \leq f(\beta_t).$$

For a smooth function, a popular choice of a surrogate function is:

$$g(\beta, \beta^-) = f(\beta^-) + < \nabla_\beta f(\beta^-), \beta - \beta^- > + \frac{\alpha}{2}\|\beta - \beta^-\|_F^2. \tag{3.9}$$

If β is replaced by β^- in Eq. (3.9), p2 holds and to satisfy p1, we should choose α in the following way:

$\alpha \geq L : \|\nabla_\beta f(\beta^1) - \nabla_\beta f(\beta^2)\|_F \leq L\|\beta^1 - \beta^2\|,$ for all β^1, β^2.

In our case, $\nabla_\beta f(\beta) = \nabla_\beta \rho_L(\beta) = \psi_L(\beta) = \frac{2\beta}{2+\beta^2}$. The Lipschitz continuity for $\psi_L(\beta)$ can be verified, since the magnitude of $\frac{2\beta}{2+\beta^2}$ is not larger than 1 for any $\beta \in \mathbb{R}$. Therefore, it follows from the mean value theorem that for any $\beta^1, \beta^2 \in \mathbb{R}$:

$\|\psi_L(\beta^1) - \psi_L(\beta^2)\|_F \leq \|\beta^1 - \beta^2\|$.

Let, $\rho_L(\mathbf{X} - \mathbf{W}) = \rho(\mathbf{W})$. Our solution strategy is the following:

- Construct a surrogate function:

$$g(\boldsymbol{W}, \boldsymbol{W}^-) = \rho(\boldsymbol{W}^-) + < \nabla_{\mathbf{W}} \rho(\boldsymbol{W}^-), \boldsymbol{W} - \boldsymbol{W}^- > + \frac{\alpha}{2}\|\mathbf{W} - \mathbf{W}^-\|_F^2 + P_0(\boldsymbol{W}) \tag{3.10}$$

$$= \alpha[\frac{1}{2}\|\mathbf{W} - \mathbf{W}^- + \frac{1}{\alpha}\nabla_{\mathbf{W}} \rho(\boldsymbol{W}^-)\|_f^2 + \frac{1}{\alpha}P_0(\boldsymbol{W})], \tag{3.11}$$

  where $\mathbf{W}^-$ is the update of $\mathbf{W}$ from the previous time point, $1/\alpha$ is the step size, and

$$P_0(\mathbf{W}) = \begin{cases} 0 & \text{if } r(\boldsymbol{W}) \leq r, \ \ \|\mathbf{W}\|_{2,0} \leq m, \\ \infty & \text{otherwise.} \end{cases}$$

- Let $\mathbf{Z} = \mathbf{W}^- - \frac{1}{\alpha}\nabla_{\mathbf{W}} \rho(\boldsymbol{W}^-)$.

- The optimization problem in Eq. (3.10) for t-th time point looks like the following:

$$\mathbf{W}_{t+1} = \underset{\mathbf{W}}{\operatorname{argmin}}\, g(\mathbf{W}, \mathbf{W}_t) \tag{3.12}$$

$$= \underset{\mathbf{W}}{\operatorname{argmin}}\, \frac{1}{2}\|\mathbf{W} - \mathbf{Z}_t\|_F^2 + \frac{1}{\alpha}P_0(\boldsymbol{W}). \tag{3.13}$$

- The problem in Eq. (3.12) looks similar to the one in the Selective PCA setup in Section 3.3. At the t-th time step, we will first calculate $\mathbf{Z}_t$ for a fixed $\mathbf{W}_t$ and then employ Algorithm 5 to update $\mathbf{V}$ and $\mathbf{S}$ correspondingly.

This strategy is detailed in Algorithm 6.

Algorithm 6: Feature Selection using Robust Loss Minimization(RLM)

Input : Rank r, $1 \leq r \leq d$, desired number of features m, maximum number of iterations M_{iter}.

Output : Set $\mathcal{I}$ of indices corresponding to the m most important features

Initialize: Initialize $\mathbf{S}, \mathbf{V}$ and $\mathbf{W}$

for $t = 1$ *to* M_{iter} **do**

Compute $\mathbf{Z}_t = \mathbf{W}_t - \frac{1}{\alpha}\nabla_{\mathbf{W}}\rho(\boldsymbol{W}_t)$

Get $\mathbf{S}_{t+1}, \mathbf{V}_{t+1}$ using Selective PCA with $\mathbf{Z}_t$ as data matrix.

Compute $\mathbf{W}_{t+1} = \mathbf{V}_{t+1}\mathbf{S}_{t+1}^T$

if $\{\|\hat{\boldsymbol{W}}_{(t+1)} - \hat{\boldsymbol{W}}_{(t)}\|_2$ is sufficiently small$\}$ break

Obtain selected set of indices corresponding to the m most important features, $\mathcal{I} = \{i : \mathbf{S}_{i\cdot} \neq \mathbf{0}\}$

Obtain the selected set of features $\mathcal{F}_m = \boldsymbol{X}[\mathcal{I}_m]$

Here we have chosen $c = 1$, assuming the data has been properly standardized. Otherwise, the estimation of scale parameter c in the Lorentzian loss function, $\rho_L(r) = \log(1 + r^2/(2c))$, can be integrated directly into the iterative robust loss minimization algorithm 6. The recommended approach is to update c at each iteration based on a robust measure of the current model's residuals. After updating the model parameters at iteration t, the residuals are computed as $\{R\}_{t+1} = \mathbf{X} - \mathbf{W}_t$. The scale for the next iteration, c_{t+1}, is then set using the square of the median of the derived residuals:

$$c_{t+1} = k \cdot (\text{median}(|\text{vec}(\mathbf{R}_{t+1})|))^2 \tag{3.14}$$

where k is a constant (e.g., $k \approx 2.2$ for consistency with Gaussian noise). This iterative refinement creates a virtuous cycle: a better model yields a more accurate estimate of the inlier noise scale, which in turn allows the robust loss to better down-weight outliers in the subsequent iteration, leading to a more stable and accurate final model.

3.5 Simulations

In this section, we conduct a simulation study to evaluate the robustness of our proposed methods against the presence of outliers in the data. The primary goal is to assess how accurately each method can perform variable selection when the dataset is contaminated with extreme values.

All datasets are generated according to the procedure described in Section 2.5.1. To introduce outliers, 2% of the samples in each generated dataset are randomly replaced with outlier data points. These outliers are simulated by drawing d-dimensional random vectors from a Cauchy distribution, characterized by its heavy tails. Specifically, we used a Cauchy distribution with a location parameter $\mu = 0$ and a scale parameter $\sigma^2 = 2$.

For each contaminated dataset, we apply the different feature selection methods. For the Signal-to-Noise Ratio (SNR) based methods (PPCA, LFA, etc.), the SNR is computed from the estimated coefficient matrix $\hat{\mathbf{W}}$ and error variance matrix $\hat{\mathbf{\Psi}}$ to rank and identify the top 10 most influential features. For the Selective PCA and our proposed Robust-Loss-based methods (RLM), the set of relevant features is directly determined from the inherent sparsity in the rows of their respective coefficient matrices.

To quantify the variable selection accuracy, we compare the set of indices of the true relevant features, denoted as $\mathcal{I}_{\text{true}}$, with the set of indices predicted as relevant by each method, denoted as $\mathcal{I}_{\text{pred}}$. The performance is then measured using the measure, described in (2.131).

We will first showcase the true feature recovery accuracy for the clean datasets (outlier-free) and then we will showcase the same for contaminated datasets.

A comparative analysis of Table 2.2 (reproduced as Table 3.1 along with Selective PCA and RLM), and Table 3.2 reveals the profound impact of outliers on the performance of different feature selection methods and unequivocally demonstrates the superior robustness of our proposed Robust-Loss based approach. Table 3.1 establishes the baseline performance in an ideal, outlier-free environment, while Table 3.2 presents the results from an identical setup but with 2% of the data contaminated by extreme values from a heavy-tailed Cauchy distribution.

In the clean data scenario, the results confirm the findings from our initial simulation. The methods designed to handle heteroscedastic noise—LFA and ELF—emerge as the clear top performers, consistently achieving the highest feature selection accuracy across nearly all conditions. LFA, in particular, demonstrates remarkable efficiency, often achieving near-perfect recovery of the 10 true signal features with only 300 to 500 samples. Our proposed Robust-Loss method, while

Table 3.1: Feature selection accuracy for outlier-free data.

Noise	n	Methods					
		PPCA	**LFA**	**ELF**	**HeteroPCA**	**Selective PCA**	**RLM**
10	50	71.2	**90.6**	87.6	84.4	73.2	72.6
	100	82.4	**97.0**	94.0	93.0	82.2	79.6
	300	88.0	**100.0**	98.0	98.8	90.2	93.6
	500	92.4	**100.0**	99.0	99.2	93.6	95.2
	1000	95.6	**100.0**	99.2	99.8	96.2	97.0
50	50	55.4	70.4	**73.4**	65.6	57.8	38.6
	100	72.6	91.8	**92.8**	87.0	71.2	57.8
	300	79.8	**100.0**	98.6	94.4	82.4	79.0
	500	86.4	**100.0**	99.4	98.6	86.8	84.6
	1000	91.4	**100.0**	99.0	99.2	91.2	95.6
100	50	49.0	57.4	**59.0**	55.8	48.2	21.6
	100	62.8	**87.4**	87.2	75.6	62.4	29.8
	300	82.0	**99.6**	**99.6**	96.4	83.2	74.6
	500	82.6	**100.0**	99.4	95.2	81.2	85.6
	1000	90.0	**100.0**	99.4	99.6	92.8	94.6

not the top performer in this ideal setting, still delivers competitive results, generally better than those of standard generative models like PPCA and HeteroPCA. This confirms that the robust loss function does not significantly compromise performance when no outliers are present.

The introduction of outliers dramatically changes the performance landscape. The performance of all standard generative methods, including the previously dominant LFA and ELF, collapses catastrophically. In the most challenging scenario (100 noise features), their accuracy plummets to barely above chance (around 10% of features). This demonstrates their extreme sensitivity to outliers; the squared-error loss inherent in their likelihood-based estimation is heavily skewed by the extreme values, leading to a complete failure in identifying the actual signal.

In stark contrast, our proposed RLM method demonstrates exceptional resilience. It consistently and overwhelmingly outperforms all other methods in every single outlier condition. Even in the most challenging scenario (100 noise features, 1000 samples), the Robust-Loss method correctly identifies 60% of the true features, whereas all other methods fail to identify more than 12%.

Table 3.2: Feature selection accuracy for data with outliers from Cauchy(0,2).

Noise	**n**	**Methods**					
		PPCA	**LFA**	**ELF**	**HeteroPCA**	**Selective PCA**	**RLM**
10	50	56.2	58.8	53.8	52.6	56.4	**64.8**
	100	55.0	57.0	53.2	53.0	55.2	**69.4**
	300	52.0	60.4	52.0	53.6	52.2	**76.0**
	500	51.6	57.6	52.6	54.4	51.4	**77.4**
	1000	52.0	55.0	51.2	51.8	52.6	**80.2**
50	50	16.0	15.5	17.0	16.0	16.5	**25.0**
	100	16.5	16.5	16.0	17.5	16.0	**33.5**
	300	19.0	14.5	17.5	16.0	18.0	**52.5**
	500	14.5	17.0	14.0	14.0	15.0	**57.0**
	1000	16.5	17.0	17.0	19.0	17.5	**60.0**
100	50	6.0	9.0	7.0	5.0	4.0	**20.0**
	100	8.0	7.0	7.0	7.0	10.0	**33.0**
	300	9.0	9.0	7.0	10.0	9.0	**43.0**
	500	8.0	11.0	8.0	11.0	8.0	**52.0**
	1000	12.0	10.0	12.0	12.0	12.0	**60.0**

CHAPTER 4

FEATURE SELECTION FOR CLASS INCREMENTAL LEARNING

4.1 Related Work

One of the foundational techniques in feature selection is Correlation-Based Feature Subset Selection (CFSS). This method evaluates and ranks subsets of features by maximizing their collective relevance to the target class while simultaneously minimizing inter-feature redundancy [60]. While effective for identifying synergistic feature groups, a key limitation of CFSS is its inability to assess the individual relevance of a feature to a specific class in a multi-class setting.

More recently, feature selection has been framed as a network pruning problem. An efficient pruning method was proposed by [39] using an l_0 sparsity constraint, allowing direct specification of the desired sparsity level. This approach iteratively removes parameters based on criteria similar to those in Feature Selection Annealing (FSA) [13]. Another powerful technique is the Thresholding-based Iterative Selection Procedure (TISP) [103, 101], which provides direct control over model sparsity by applying a thresholding function to network parameters. A related approach, inspired by deep networks, was proposed by [7], in which autoencoders are first used to extract features, and a pruning algorithm then constructs a subset by minimizing the input reconstruction error. While powerful, these methods' selection criteria are based on sparsity or reconstruction error rather than a direct, model-based measure of a feature's discriminatory signal for a particular class.

Principal Component Analysis (PCA) has been widely adopted for supervised feature selection. Supervised PCA, introduced by [120], incorporates class labels to identify principal components that capture both high variance and strong class separation. However, this method faces significant scalability challenges, as the entire model must be retrained whenever a new data class is introduced.

To address this, recent work [114] has explored the use of Probabilistic PCA (PPCA) for multi-class classification. Their approach models each class separately, enabling class-incremental learning, but their work did not include a feature selection mechanism.

Hybrid methods, such as the PCA-Logistic regression framework used by [127] for facial recognition, have also been proposed. A significant drawback of such approaches is their reliance on

accessing the entire dataset for dimensionality reduction. More importantly, these methods perform feature transformation (or extraction), not selection. The resulting principal components are linear combinations of all original features, meaning that even when using a reduced set of components, one must still measure every original feature, which can be impractical and harms interpretability.

Our proposed method offers a distinct and practical alternative to the techniques reviewed above. It is founded on three core principles: class-specific modeling and feature selection via a Signal-to-Noise Ratio (SNR) criterion, and a unique classification mechanism based on the Mahalanobis distance over class-specific feature sets.

To classify a new observation, we compute its Mahalanobis distance to each class. Crucially, the distance to a given class j is calculated only using the feature subset selected for that class j. This means that for a single new data point, the classification process involves projecting it onto multiple, distinct feature subspaces—one for each potential class—and evaluating its distance within each subspace, accounting for that class's unique covariance structure.

This approach represents a fundamental departure from feature transformation methods. Our method performs true feature selection, identifying a parsimonious subset of original features for each class. This is a critical advantage over PCA-based techniques, which create new features from linear combinations of all original variables. By not relying on any transformation, our method ensures that only the selected features need to be measured for classification, leading to a truly reduced and interpretable model. This makes it highly efficient and practical for real-world applications where data acquisition is costly.

4.2 Multi-class Classification

After selecting the relevant features, we next conduct multi-class classification using them. For each class, we will individually train the model, estimate the parameters, perform feature selection based on these estimates, and store the results. When predicting the class for a new observation, we will compute the posterior probability that the new observation belongs to each class using the selected features, ultimately assigning the new observation to the class with the maximum likelihood.

First, we will define certain variables before proceeding to technical details in this section. Y represents the class label ($Y \in \{1, 2, 3, \cdots, C\}$). Let m and n denote the number of selected features

for every class and the total number of training samples from all the classes, respectively, and n_i be the number of training observations in the i^{th} class. The random vector corresponding to the set of all features is denoted as $\mathbf{x} = (x_1, x_2, \cdots, x_d)$, whereas $\mathbf{x}^{(j)}$ and $\mathbf{x}^{(-j)}$ represents the random vector corresponding to selected set of feature and the remaining features for class j. Let $\mathcal{J}$ denote the m indices of selected features for class j, i.e. $\mathcal{J} \subset \{1, 2, \cdots, d\}$. Therefore, $\mathbf{x}^{(j)} = \mathbf{x}[\mathcal{J}]$ and $x^{(-j)} = x[\{1, 2, \cdots, d\} \cap \overline{\mathcal{J}}]$.

4.2.1 PPCA

In this section, to perform multi-class classification, we will calculate the probability score associated with every class. For class j, the selected set of features will follow a normal distribution:

$$\mathbf{x}^{(j)} \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{4.1}$$

$$\boldsymbol{\Sigma}_j = W^{(j)} W^{(j)T} + \sigma_j^2 \mathbf{I}_m, \text{ and} \tag{4.2}$$

$$\mathbf{x}^{(-j)} \sim N(\boldsymbol{\mu}_{-j}, \sigma_j^2 \mathbf{I}_{d-m}). \tag{4.3}$$

Here, $\mathbf{x}^{(j)}$ and $\mathbf{x}^{(-j)}$ are independent from each other. $\hat{\mathbf{W}^{(j)}}$ and $\hat{\sigma}_j^2$ can be achieved from the closed form of ML estimates, detailed in equation (2.5).

The following lemma describes how to make class predictions for a new observation $\mathbf{x}^{new}$ using PPCA.

Lemma 6. *For a new set of features, $\mathbf{x}^{new}$, we will assign it to class k,if,*

$$k = \underset{j \in \{1,2,\cdots,c\}}{\operatorname{argmin}} \{ \frac{\mathbf{S}_j(\mathbf{x}^{new}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{2} + \frac{\mathbf{E}_j^{PPCA}(\mathbf{x}^{new})}{2} + a_j \} \tag{4.4}$$

where $\mathbf{S}_j(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = (\mathbf{x}^{(j)} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}^{(j)} - \boldsymbol{\mu}_j)$, $\mathbf{E}_j^{PPCA}(\mathbf{x}) = \frac{(\mathbf{x}^{(-j)} - \boldsymbol{\mu}_{-j})^T (\mathbf{x}^{(-j)} - \boldsymbol{\mu}_{-j})}{\sigma_j^2}$ *and* $a_j = \frac{1}{2} * (\log |\boldsymbol{\Sigma}_j| + (d - m) \log(\sigma_j^2)) - \log n_j$.

Proof. For a new set of features, $\mathbf{x}^{new}$, we will assign it to class k, if:

$$
\begin{aligned}
k &= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} P(Y=j|\mathbf{x}=\mathbf{x}^{new}) \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} \frac{P(\mathbf{x}=\mathbf{x}^{new}|Y=j)P(Y=j)}{P(\mathbf{x}=\mathbf{x}^{new})} \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} P(\mathbf{x}=\mathbf{x}^{new}|Y=j)P(Y=j) \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} P(\mathbf{x}^{(j)}=\mathbf{x}^{new(j)})P(\mathbf{x}^{(-j)}=\mathbf{x}^{new(-j)})P(Y=j) \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmin}} \{\frac{1}{2}(\log|\mathbf{\Sigma}_j| + (\mathbf{x}^{new(j)}-\boldsymbol{\mu}_j)^T\mathbf{\Sigma}_j^{-1}(\mathbf{x}^{new(j)}-\boldsymbol{\mu}_j) + (d-m)\log(\sigma_j^2) + \\
&\qquad\qquad \frac{(\mathbf{x}^{new(-j)}-\boldsymbol{\mu}_{-j})^T(\mathbf{x}^{new(-j)}-\boldsymbol{\mu}_{-j})}{\sigma_j^2}) - \log(\frac{n_j}{n})\} \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmin}} \{\frac{\mathbf{S}_j(\mathbf{x}^{new},\boldsymbol{\mu}_j,\mathbf{\Sigma}_j)}{2} + \frac{\mathbf{E}_j^{PPCA}(\mathbf{x}^{new})}{2} + a_j\}
\end{aligned}
$$

and $\log n$ is a constant which can be omitted. Here $P(Y=j)$ has been approximated by utilizing the ratio of sample observations n_j relative to the total number of observations n i.e. $P(Y=j) = \frac{n_j}{n}$. □

If the the number m of selected features is large (e.g., $m = 4096$), the computation of $S_j(\mathbf{x})$, for each observation, involves multiplication with a large $m \times m$ matrix, which can be expensive.

Hence, to simplify the computational burden in equation (4.4), an alternative theorem will be employed. In this scenario, we will perform SVD on the estimated covariance matrix $\hat{\mathbf{\Sigma}}_j^0$ to obtain the rank-r estimate of it.

$$\hat{\mathbf{\Sigma}}_j^0 = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{4.5}$$

$$\hat{\mathbf{\Sigma}}_j = \mathbf{L}_j\mathbf{D}_j\mathbf{L}_j^T + \lambda I_m, \tag{4.6}$$

where λ has been considered as 0.01, $\mathbf{L}_j(\in \mathbb{R}^{m\times r}, r << m)$ consists of first r columns of $\mathbf{V}$ and $\mathbf{D}_j$ is the diagonal matrix with r largest singular values from $\mathbf{D}$.

Now we will define the alternative score variable denoted as $r_j(\mathbf{x})$, and subsequently present a theorem that employs the same variable.

$$r_j(\mathbf{x}) = r(\mathbf{x}^{(j)};\boldsymbol{\mu}_j,\mathbf{L}_j,\mathbf{D}_j) = \|\mathbf{x}^{(j)}-\boldsymbol{\mu}_j\|_2^2/\lambda - \|\mathbf{u}(\mathbf{x}^{(j)})\|_2^2/\lambda \tag{4.7}$$

where, $\mathbf{u}(\mathbf{x}) = \operatorname{diag}(\frac{\sqrt{\mathbf{d}_j}}{\sqrt{\mathbf{d}_j+\lambda 1_r}})\mathbf{L}_j^T(\mathbf{x}-\boldsymbol{\mu}_j)$ and $\mathbf{d}_j \in \mathbb{R}^r$ is the vector consisting of the diagonal elements of $\mathbf{D}_j$. The tall matrix $\mathbf{L}_j$, makes the matrix multiplication procedure less time consuming.

Theorem 13. *(due to [114]) The score variable in Lemma 6 can also be written as:* $S_j(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = r_j(\mathbf{x})$*, and* $\log|\boldsymbol{\Sigma}_j| = (m-r)\log\lambda + \sum_{l=1}^{r}(\mathbf{d}_j[l] + \lambda)$*, where,* $\mathbf{d}_j[l]$ *denotes the* l^{th} *element of* $\mathbf{d}_j$*.*

4.2.2 LFA

Multi-class classification using LFA is similar to PPCA. The only difference being the involvement of $\boldsymbol{\Psi}_j$ instead of $\sigma_j^2\mathbf{I}_d$ in the probability score for the j^{th} class.

Similar to PPCA, using the model in eq.(2.1), the distribution of $\mathbf{x}^{(i)}$ is:

$$\mathbf{x}^{(j)} \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}'_j) \tag{4.8}$$

$$\boldsymbol{\Sigma}'_j = \mathbf{W}^{(j)}\mathbf{W}^{(j)T} + \boldsymbol{\Psi}^{(j)} \tag{4.9}$$

$$\mathbf{x}^{(-j)} \sim N(\boldsymbol{\mu}_{-j}, \boldsymbol{\Psi}^{(-j)}). \tag{4.10}$$

The estimated $\hat{\mathbf{W}^{(j)}}$ and $\hat{\boldsymbol{\Psi}}^{(j)}$, and $\hat{\boldsymbol{\Psi}}^{(-j)}$ can be obtained after convergence of the EM algorithm, in Theorem 1.

Lemma 7. *For a new observation* $\mathbf{x}^{new}$*, we will assign it to class* k *with*

$$k = \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmin}} \frac{S_j(\mathbf{x}^{new}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}'_j)}{2} + \frac{\mathbf{E}_j^{LFA}(\mathbf{x}^{new})}{2} + b_j \tag{4.11}$$

where $S_j(\mathbf{x}^{new}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}'_j)$ *has been defined in Lemma 6,*
$\mathbf{E}_j^{LFA}(\mathbf{x}) = (\mathbf{x}^{new(-j)} - \boldsymbol{\mu}_{-j})^T(\boldsymbol{\Psi}^{(-j)})^{-1}(\mathbf{x}^{new(-j)} - \boldsymbol{\mu}_{-j})$ *and*

$$b_j = \frac{1}{2}(\log|\boldsymbol{\Sigma}'_j| + \log|\boldsymbol{\Psi}^{(-j)}|) - \log n_j. \tag{4.12}$$

Proof. Using Lemma 6, we get:

$$\begin{aligned}
k &= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} P(\mathbf{x}^{(j)} = \mathbf{x}^{new(j)})P(Y=j) \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmax}} P(\mathbf{x}^{(j)} = \mathbf{x}^{new(j)})P(\mathbf{x}^{(-j)} = \mathbf{x}^{new(-j)})P(Y=j) \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmin}} [\frac{1}{2}(\log|\boldsymbol{\Sigma}'_j| + (\mathbf{x}^{new(j)} - \boldsymbol{\mu}_j)^T(\boldsymbol{\Sigma}'_j)^{-1}(\mathbf{x}^{new(j)} - \boldsymbol{\mu}_j) \\
&\qquad + (\mathbf{x}^{new(-j)} - \boldsymbol{\mu}_{-j})^T(\boldsymbol{\Psi}^{(-j)})^{-1}(\mathbf{x}^{new(-j)} - \boldsymbol{\mu}_{-j}) + \log|\boldsymbol{\Psi}^{(-j)}|) - \log n_j] \\
&= \underset{j\in\{1,2,\cdots,c\}}{\operatorname{argmin}} \frac{S_j(\mathbf{x}^{new}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}'_j)}{2} + \frac{\mathbf{E}_j^{LFA}(\mathbf{x}^{new})}{2} + b_j.
\end{aligned}$$

□

4.2.3 Unified Approach

In this section, we propose a unified approach for multi-class classification to compare model performance. We will only employ the selected features for a new observation, $\mathbf{x}^{new(j)}$, and calculate the Mahalanobis distance over all the given classes. It is defined as the following:

$$MD(\mathbf{x}^{new}, class_j) = (\mathbf{x}^{new(j)} - \hat{\boldsymbol{\mu}}_j)^T \hat{\boldsymbol{\Sigma}}_j^{-1} (\mathbf{x}^{new(j)} - \hat{\boldsymbol{\mu}}_j) \tag{4.13}$$

with $\hat{\mu}_j$ and $\hat{\boldsymbol{\Sigma}}_j$ are the sample mean and estimated covariance of the signals for class j, respectively. There are a few advantages of using the Mahalanobis distance over the Frobenius norm:

- In the previous section, we have seen that the estimated posterior probability corresponding to class prediction k, $P(\mathbf{Y} = k|\mathbf{x} = \mathbf{x}^{new})$, yields a minimum in $MD(\mathbf{x}^{new}, class_j) + E_j^{PPCA} + a_j$ for PPCA and $MD(\mathbf{x}^{new}, class_j) + E_j^{LFA} + b_j$, for LFA. Here, (a_j, b_j) are constants (do not depend on $\mathbf{x}^{new}$) and E_j^{PPCA}, E_j^{LFA} attributed to noise variables for class j and do not have meaningful contribution to this equation. Hence, it is rational to compute the Mahalanobis distance and exclude E_j^{PPCA} and E_j^{LFA}, while focusing solely on relevant features for multi-class classification.

- In high-dimensional scenarios, the Mahalanobis distance is preferred over the Frobenius norm because it considers the covariance structure of the data. The Frobenius norm may struggle to accurately capture variable relationships in high-dimensional spaces accurately, whereas the Mahalanobis distance normalizes differences by variances and covariances, making it more robust. This enhances its accuracy in task classification.

To classify a new observation($\mathbf{x}^{new}$), our unified approach consists of two steps:

- Calculate the score for every class j: $MD_j = MD(\mathbf{x}^{new}, class_j)$.

- Predicted class(k) will be the one that results in a minimum value of the score variable, $k = \text{argmin}_{j \in \{1,2,\cdots,C\}} MD_j$

When dealing with multiclass classification, where the feature count (m) is high, computing $MD(\mathbf{x}^{new}, class_j)$ becomes time-consuming due to the matrix multiplication of size $m \times m$. To overcome this computational burden, we can substitute it with the r-score $r_j(\mathbf{x}^{new})$.

For PPCA, we assume the same noise variance over all the dimensions. SRRR and Robust loss minimization seek a low-rank, sparse representation while screening essential variables. Therefore, these methods can be viewed as non-parametric versions of PPCA with an added sparsity constraint. For these three methods, we can directly replace $MD(\mathbf{x}^{new}, class_j)$, with $r_j(\mathbf{x}^{new})$.

LFA, ELF, and HeteroPCA are the methods based on the latent factor model. They share the assumption that the noise covariance structure ($\mathbf{\Psi}$) is not isotropic. Therefore, the PPCA r-score (3.4) is not valid in such cases. Here, we present a general version of the PPCA r-score that reduces the computation time of the MD-score while using LFA models for feature selection in multi-class classification.

To make use of Theorem 13 for LFA models with non-isotropic noise variance $\mathbf{\Psi}$, we employ the following theorem to make the Mahalanobis distance computation faster.

Theorem 14. *If*

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{D}\mathbf{L}^T + \mathbf{\Psi}, \tag{4.14}$$

with $\mathbf{L} \in \mathbb{R}^{m\times r}$*, and diagonal matrices* $\mathbf{D} \in \mathbb{R}^{r\times r}$ *and* $\mathbf{\Psi} \in \mathbb{R}^{m\times m}$ *with positive entries, the Mahalanobis distance can be computed as:*

$$MD(\mathbf{x}, \boldsymbol{\mu}, \mathbf{\Sigma}) = r(\mathbf{\Psi}^{-\frac{1}{2}}\mathbf{x}; \mathbf{\Psi}^{-\frac{1}{2}}\boldsymbol{\mu}, \mathbf{L}', \mathbf{D}', 1) \tag{4.15}$$

where $r(\mathbf{x}; \boldsymbol{\mu}, \mathbf{L}, \mathbf{D}, \lambda)$ *is defined in* (4.7)*, and* $\mathbf{L}'$ *and* $\mathbf{D}'$ *are obtained by SVD on* $\mathbf{\Sigma}' = \mathbf{\Psi}^{-\frac{1}{2}}\mathbf{\Sigma}\mathbf{\Psi}^{-\frac{1}{2}}$*.*

Proof. We consider the following transformation:

$$\begin{aligned}
\mathbf{x}' &= \mathbf{\Psi}^{-\frac{1}{2}}\mathbf{x}, \\
\boldsymbol{\mu}' &= \mathbf{\Psi}^{-\frac{1}{2}}\mu, \\
\mathbf{\Sigma}' &= (\mathbf{\Psi}^{-\frac{1}{2}}\mathbf{W})(\mathbf{\Psi}^{-\frac{1}{2}}\mathbf{W}^T) + \mathbf{I}_m.
\end{aligned}$$

$\mathbf{\Sigma}'$ looks similar to (4.1), with $\lambda = 1$. Therefore, using the Proposition 13, we get: $MD(\mathbf{x}', \boldsymbol{\mu}', \mathbf{\Sigma}') = r(\mathbf{x}'; \boldsymbol{\mu}', \mathbf{L}', \mathbf{D}', 1)$. Here, $\mathbf{\Sigma}' = \mathbf{\Psi}^{-\frac{1}{2}}\mathbf{\Sigma}\mathbf{\Psi}^{-\frac{1}{2}} = \mathbf{L}'\mathbf{D}'\mathbf{L}'^T$. Also,

$$\begin{aligned}
MD(\mathbf{x}, \boldsymbol{\mu}, \mathbf{\Sigma}) &= (\mathbf{x} - \boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \\
&= (\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{\Psi}^{-\frac{1}{2}})(\mathbf{\Psi}^{\frac{1}{2}})\mathbf{\Sigma}^{-1}(\mathbf{\Psi}^{\frac{1}{2}})(\mathbf{\Psi}^{-\frac{1}{2}})(\mathbf{x} - \boldsymbol{\mu}) \\
&= (\mathbf{x}' - \boldsymbol{\mu}')^T(\mathbf{\Psi}^{-\frac{1}{2}}\mathbf{\Sigma}\mathbf{\Psi}^{-\frac{1}{2}})^{-1}(\mathbf{x}' - \boldsymbol{\mu}') \\
&= (\mathbf{x}' - \boldsymbol{\mu}')^T\mathbf{\Sigma}'^{-1}(\mathbf{x}' - \boldsymbol{\mu}') \\
&= MD(\mathbf{x}', \boldsymbol{\mu}', \mathbf{\Sigma}') \\
&= r(\mathbf{x}'; \boldsymbol{\mu}', \mathbf{L}', \mathbf{D}', 1) \\
&= r(\mathbf{\Psi}^{-\frac{1}{2}}\mathbf{x}; \mathbf{\Psi}^{-\frac{1}{2}}\boldsymbol{\mu}, \mathbf{L}', \mathbf{D}', 1). \ \square
\end{aligned}$$

Therefore, when incorporating the r-score into the computation of classification scores for these methods, the following steps are performed:

1. Consider the transformation: $\mathbf{u}^{(j)} = \hat{\mathbf{\Psi}}^{(j)-\frac{1}{2}}\mathbf{x}^{(j)}$

2. Calculate the score for every class j: $MD(\mathbf{x}^{new}, class_j) = MD(\mathbf{u}^{new}, class_j) = r_j(\mathbf{u}^{new})$.

4.3 Class Incremental Learning

4.3.1 Introduction

Class Incremental learning is a machine learning paradigm that enables a model to continuously learn from new data without requiring retraining on the entire dataset from scratch. This approach is essential for large-scale, dynamic environments where new classes or tasks are introduced sequentially over time. The primary goal is to update an existing model to incorporate new information while retaining the knowledge it has already acquired from previous data. It is the process of adding new classes to a model without losing the understanding of previously learned classes, thereby facilitating learning on massive datasets with lower computational and memory costs. This paradigm is crucial for building scalable systems that can adapt and evolve without the prohibitive expense of repeated, full-scale training sessions.

Catastrophic forgetting is the primary challenge in incremental learning, a phenomenon in which a model's performance on previously learned tasks drastically deteriorates after it is trained on a new task. When a neural network is fine-tuned on new classes, its parameters (weights) are adjusted to minimize the error for those new classes. This optimization process often overwrites or interferes with the parameter configurations that were essential for correctly classifying the old courses. This is a particular problem for conventional classifiers, leading to a severe drop in accuracy for the old classes.

4.3.2 Literature Review

The literature has explored two primary schools of thought to mitigate this issue: (1) methods that aim to preserve the knowledge of old classes, typically through regularization or the use of exemplars, and (2) methods that focus on correcting the inherent bias towards new classes that emerges in the classifier.

One of the popular works for regularization based approaches includes knowledge distillation(KD) application by [46]. KD preserves a model's prior behavior by adding a loss term that minimizes the difference between the softened output probabilities of the current model and a stored (or previous) version of itself on old classes—typically using temperature-scaled softmax outputs.

This regularization encourages stability in decision boundaries without requiring raw old data. The following methods, reviewed chronologically, leverage KD or alternative approaches to address forgetting. Early and influential approaches to incremental learning focused on preserving the learned feature representations. [121] proposed paying attention to specific activation maps to distill knowledge from teacher to student models effectively. This approach laid the foundation for subsequent works that incorporate attention mechanisms to prevent changes in feature representations during continual learning. Building on these ideas, the Incremental Classifier and Representation Learning (iCaRL) framework, introduced by [89], dynamically updates these exemplars after each training stage and employs a nearest-class-mean classifier along with knowledge distillation loss to mitigate forgetting. Additionally, iCaRL integrates task-specific parameters and builds a mechanism to store representative samples, ensuring balanced performance across old and new classes. [28] employs an attention distillation loss to transfer knowledge without needing data from base classes. To maintain performance on prior tasks, they use a gradient that incorporates information which does not change features of old classes significantly. [30] introduced a multi-scale feature distillation strategy that applies knowledge distillation to pooled outputs at different spatial resolutions in a CNN. By enforcing consistency across feature map levels, it preserves spatial and semantic information from old tasks. It achieves strong performance in exemplar-free and low-exemplar regimes, outperforming iCaRL in several benchmarks.

While the aforementioned methods focus on preserving the feature extractor, another line of research focuses on a strong bias towards the most recently seen classes. [47] identified the imbalance between old and new classes as a primary cause of catastrophic forgetting and proposed UCIR to address task recency bias. They replace the standard softmax layer with a cosine normalization layer to mitigate this imbalance. This bias-correction method aims to create a unified classifier that performs well across all classes by rebalancing the learning process. [118] discovered a strong bias towards new classes in the last fully connected layer of CNNs and introduced BiC to correct this task bias. Their method adds an additional layer for bias correction and divides training into two stages: one for model training and another using a validation set to estimate and adjust the bias.

Some of the modern approaches employ pre-trained Models(PTM). [81] proposed a k-Nearest Neighbor (KNN) classifier based on CLIP[86]. This method evaluates the classifier on several popular benchmarks and achieves state-of-the-art performance in continual learning settings. By leveraging CLIP's robust features, it dynamically handles old and new classes without severe forget-

ting. Following a similar philosophy, [114] also proposed using a frozen, pre-trained self-supervised feature extractor to ensure feature consistency across all incremental tasks. However, instead of a non-parametric classifier, they train a separate, generative Probabilistic Principal Component Analysis (PPCA) model for each class and structurally prevent catastrophic forgetting. This domain of PTM based research in class incremental learning is gaining popularity. PTM based works utilize CLIP or Dinov2 as the backbone to extract deep features from images and incorporate prototype classifiers [128], knowledge rumination[33] to enhance generalization and adaptability while keeping the amount of catastrophic forgetting minimal in the class incremental setup.

4.3.3 A Generative and Feature-Selective Approach to CIL

Our proposed framework aligns with the modern school of thought in Class-Incremental Learning, leveraging powerful pre-trained Models (PTMs) to provide a stable feature space. However, it introduces a critical and novel extension by integrating a class-specific feature selection mechanism. Building on the philosophy of works such as [114], which use a frozen feature extractor and separate generative models per class, our approach not only learns a unique distribution for each class but also identifies the most salient original features necessary for its recognition. This architecture offers a fundamental and structural solution to catastrophic forgetting, contrasting sharply with the preservation and bias-correction methods discussed previously. Unlike regularization-based approaches [46, 89] that require exemplars or knowledge distillation to approximate past knowledge, our method perfectly preserves it. It also sidesteps the need for complex bias-correction layers [47, 118] that re-balance a shared classifier. The suitability of our framework for CIL is rooted in its class-specific design for the following reasons:

- **Structural Immunity to Forgetting:** The core challenge of catastrophic forgetting arises from the overwriting of shared parameters in a monolithic model. By dedicating a separate, independent generative model to each class, we eliminate this issue by design. The parameters learned for a new class have no architectural pathway to interfere with the parameters of previously learned classes. Preserving these learned parameters ensures prior knowledge is fully retained, not just approximated.
- **Constant-Time Model Expansion:** Traditional classifiers often require retraining on a growing dataset, with complexity scaling as $O(C)$ with the number of classes C. Our approach is significantly more scalable. The model for each class is "wrapped" tightly around its own observations. When a new class is introduced, we train a new, independent model for it. This allows the system to expand its knowledge with a computational cost that scales as $O(1)$ per new class, making it ideal for dynamic, large-scale environments.

- **Adaptive and Interpretable Feature Selection:** The key novelty of our extension is the integration of Signal-to-Noise Ratio (SNR)-based feature selection. As each new class model is trained, we also identify the most discriminative subset of features from the PTM's output space for that specific class. This ensures that the system learns not only what a new class looks like (its distribution) but also which features matter most for identifying it. This adds a layer of adaptability and interpretability that is not present in prior generative CIL work.

While a comprehensive methodological exploration of CIL is beyond the primary scope of this dissertation, the inherent properties of our framework make it a powerful and elegant solution to the CIL problem. We apply it in a CIL setting primarily to empirically validate its structural robustness against catastrophic forgetting, with a detailed comparative analysis presented in the results section.

4.4 Real Data Experiments

We evaluate the proposed feature selection for multi-class classification methods on three widely utilized popular image classification datasets: CIFAR-10 [65], CIFAR-100 [66] and ImageNet-1k [94].

4.4.1 ImageNet-1k

The ImageNet-1k dataset, formally known as the dataset for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) classification task, stands as one of the most influential benchmarks in the history of computer vision [93]. The "1k" designation refers to its core task: classifying images into one of 1000 distinct object categories. The dataset is massive, comprising approximately 1.28 million high-resolution images for training, 50,000 for validation, and 100,000 for testing.

ImageNet-1k's importance is immense; it sparked the deep learning revolution. AlexNet's success at ILSVRC 2012 showcased deep CNNs' capabilities and influenced two decades of research [67]. Its scale and complexity made ImageNet-1k the benchmark in image classification. Generally, good performance on it is considered a model's ability to learn robust, generalizable visual features [98, 45, 29, 75].

From a statistical and machine learning perspective, the primary challenges of ImageNet-1k stem from its class structure and data properties. These create two key difficulties: high intra-class variance, where instances of the same class (e.g., chair) can appear in vastly different poses,

lighting conditions, etc., and low inter-class variance, where different classes (e.g., Siberian husky vs. Eskimo dog) can be almost visually indistinguishable. An effective model must learn representations invariant to diverse appearances within a class while sensitive to subtle differences distinguishing closely related classes. ImageNet-1k serves as a key testbed for scalability and robustness. Its large scale tests the efficiency of our feature selection framework, and its fine-grained classes evaluate our SNR-based criterion's ability to identify subtle, discriminative features crucial for accurate classification.

4.4.2 CIFAR 10/100

The CIFAR-10 and CIFAR-100 datasets are cornerstone benchmarks in the field of computer vision and image classification [65, 66]. Both datasets consist of 60,000 32x32 pixel color (RGB) images, which are partitioned into a standard training set of 50,000 images and a test set of 10,000 images. Due to their manageable size and well-defined structure, they have become ubiquitous in the machine learning literature [125, 100] and serve as a standard testbed for evaluating the ability to learn meaningful representations from low-resolution data [23, 55, 90, 2, 64].

The primary distinction between the two datasets lies in their class granularity and hierarchical structure. CIFAR-10 divides its images into 10 coarse, mutually exclusive object classes: "airplane", "automobile", "bird", "cat",etc. This presents a general object recognition task where the categories are semantically distinct. In contrast, CIFAR-100 presents a more significant challenge in fine-grained classification. It contains 100 distinct classes, which are further grouped into 20 superclasses. For example, the superclass "trees" contains subclasses such as "oak tree", "maple tree", "pine tree", while the superclass "aquatic mammals" includes "beaver", "dolphin", etc. This hierarchical structure makes CIFAR-100 an excellent benchmark for testing a model's ability to learn nuanced and detailed feature representations.

In the context of this dissertation, the CIFAR allows us to evaluate our feature selection framework on tasks of varying classification complexity (10 vs. 100 classes) before scaling up to ImageNet. The primary challenge posed by these datasets is their low $32x32$ resolution. This forces the feature extractor to produce representations from highly constrained input and provides a stringent test of our SNR-based method's ability to identify the most salient and robust features from a potentially noisy, low-information signal. Therefore, strong performance on CIFAR-100, in particular, demonstrates the framework's effectiveness in a fine-grained, low-resolution setting, a common and challenging scenario in practical applications.

4.4.3 Deep Feature Extractors for Images

The success of modern machine learning systems is critically dependent on the quality of the underlying feature representations. From a statistical perspective, raw image data, represented as a grid of pixels, presents a formidable challenge for direct modeling due to its high dimensionality and complex, non-linear dependencies. Deep learning models provide a principled, data-driven solution to this representation learning problem [38]. These networks learn a hierarchical transformation, $\Phi(\cdot)$, mapping the raw pixel space into a high-level vector space where semantic relationships are more explicit. By leveraging models pre-trained on massive datasets, we engage in a form of large-scale transfer learning, using "off-the-shelf" features that have been proven to be highly generalizable across a wide variety of downstream tasks [98]. This paradigm has been the driving force behind the success for variety of vision related tasks [45, 107, 129]. Therefore, our approach of extracting deep features is a principled method to obtain a state-of-the-art data representation upon which our statistical feature selection techniques can be most effectively applied. Specifically, we utilize two of the most influential models: Contrastive Language-Image Pre-training (CLIP) [86] and a self-supervised Vision Transformer (DINOv3) [105].

CLIP. Contrastive Language-Image Pre-training (CLIP)[86] consists of two parallel encoders: an image encoder (typically a Vision Transformer or a large ResNet) and a text encoder (a standard text Transformer). The model was trained on a massive, web-scale dataset of 400 million image-text pairs. The training objective is a contrastive loss: for a given batch of images and texts, the model learns to maximize the similarity between the embeddings of correct image-text pairs. Its immense popularity stems from its remarkable ability to classify images into categories it was not explicitly trained on, by leveraging the natural language descriptions of those categories[87]. The resulting image representations from CLIP consists of deep semantic meaning and are extremely useful for vision related tasks [114, 81, 43].

The image CNN component of CLIP incorporates a prominent attention mechanism as its final layer before the classification layer. For our purposes, we utilized a pretrained modified ResNet-50 classifier known as RN50x4 from the CLIP GitHub package [86]. The CLIP feature extractor is trained with medium resolution 288×288 images. Therefore, prior to processing, input images were resized to 288×288 for the ImaeNet-1k dataset. For CIFAR-10/100, we resize the original images to 144×144. These images, when resized to 288×288, they will look very blurred. [114] showed that the 144×144 input is the best setting for low resolution images from CIFAR-100 for CLIP

feature extractor. Therefore, we have utilized the same resizing factor, 144 in our setup as well. The extracted numerical features for ImageNet-1k is 640 dimensional and for CIFAR datasets, the feature dimension is 2560.

DINOv3. Similarly, we employ DINOv3 as a second, recent and philosophically distinct feature extractor, representing the state-of-the-art in self-supervised visual representation learning [105]. The model architecture is a large-scale Vision Transformer (ViT-L) that learns from a massive, curated dataset of 1.7 billion images without any textual labels. Its training objective is based on a sophisticated self-distillation and image token matching process within a student-teacher framework. This forces the model to learn representations that are invariant to augmentations and capture the fine-grained structure of visual content. This model has gained popularity in a short period of time. It is being mostly utilized in areas like object detection [104], medical vision [73], etc.

For feature extraction, we employ the powerful vitl16 model, pre-trained on the LVD−1.7B dataset. The model is loaded into memory using the Hugging Face transformers library, which ensures access to the official, pre-trained weights from the Meta AI repository.

Each image is first resized so that its shorter edge is 512 pixels, and then, a 512×512 pixel patch is extracted from the center of the resized image. We extract the final, high-level feature representation for these images, from the pooler output of the model. This output corresponds to the 1024−dimensional embedding after it has been processed by the final layers of the transformer, serving as a holistic representation of the entire image's content.

Discussion. The core difference between CLIP and Dinov3 is their supervisory signals. CLIP uses weak supervision from natural language, learning to map visual objects to concepts like "dog". DINOv3's internal supervision recognizes objects by matching parts across views, understanding what a "dog" looks like. Therefore, CLIP captures high-level semantics, while DINOv3 focuses on detailed visual structure. Testing our feature selection on both models shows its effectiveness across these distinct paradigms.

We would also like to discuss the training sets of CLIP and Dinov3 because our results on ImageNet-1k and CIFAR can be less reliable if there is an overlapping between the training sets and ImageNet. In both papers, they mentioned that their datasets are created from a variety of publicly available sources on the Internet for CLIP and a large data pool of web images collected from public posts on Instagram. Although the train split of ImageNet-1k is used for Dinov3 training,

they do not employ the validation set for the same. More convincingly, both CLIP and Dinov3 have been evaluated on ImageNet-1k in their respective papers. In addition, an existing method applied CLIP to CIL [43]. After considering the above evidence, using a CLIP-based encoder on ImageNet benchmarks is reasonable.

4.4.4 Models for Comparison

We compare the feature selection efficiency of the proposed methods against two popular methods, Feature Selection with Annealing (FSA) [13] and TISP [101] with soft thresholding (L1 penalty), applied on the same data (features) as the other methods. FSA and TISP were implemented as a fully connected one-layer neural network with cross-entropy loss. The models were trained for 30 epochs using the Adam optimizer[61] (learning rate: 0.001).

Feature Selection Annealing(FSA). Feature Selection with Annealing (FSA) is an embedded feature selection method that integrates sparsity enforcement directly into the iterative optimization process [13]. For a standard regression problem of the form $\mathbf{y} = \mathbf{X}\mathbf{b} + \boldsymbol{\epsilon}$, FSA aims to find a sparse estimate of the coefficient vector $\mathbf{b}$. The method operates by alternating between two steps at each epoch e:

$$\begin{aligned} \mathbf{b} &= \mathbf{b} - \eta \frac{\partial \ell(\mathbf{X}, \mathbf{y}, \mathbf{b})}{\partial \mathbf{b}} \\ m^e &= k + (d-k)\max\left(0, \frac{\text{epochs} - 2e}{2e\mu + \text{epochs}}\right). \text{ Keep only } m^e \text{ variables with highest } |b_j| \end{aligned} \tag{4.16}$$

Here k denotes the number of selected features, μ controls the convexity of the schedule, allowing it to range from a linear decay ($\mu = 0$) to a rapid, non-linear drop ($\mu > 0$). m^e is defined as the annealing schedule, which specifies the exact number of non-zero features to be retained at epoch e. After the gradient update, the algorithm keeps only the m^e coefficients with the largest magnitudes and sets all others to zero. The schedule is designed to gradually reduce the number of active features from the total number d of features down to a desired final number, k. A key aspect of FSA is the use of non-linear schedules that drop features aggressively in early epochs and more slowly in later ones, a strategy designed to quickly eliminate irrelevant predictors while allowing for more careful estimation of the remaining, more ambiguous features.

Thresholding-based Iterative Selection Procedure (TISP). The Thresholding-based Iterative Selection Procedure (TISP) is a general and efficient algorithmic framework for solving

a wide class of penalized optimization problems, making it a powerful tool for embedded feature selection [103, 101]. The core of the method is to find a sparse coefficient vector $\mathbf{b}$ that minimizes a composite objective function, which combines a data fidelity term with a sparsity-inducing penalty term:

$$L(\mathbf{b}) = \frac{1}{N}\sum_{j=1}^{N}\mathcal{L}_{\text{data}}(\mathbf{b}^T\mathbf{x}_j, y_j) + \lambda\sum_{i=1}^{p}P(b_i), \tag{4.17}$$

where $\mathcal{L}_{\text{data}}$ is a loss function such as squared error or logistic loss, and P is a θ induced penalty function. In this case, $\mathcal{L}_{\text{data}}$ is a cross-entropy loss.

TISP operates by iterating between two simple steps: a standard gradient descent step on the data loss term, followed by the application of a thresholding operator $\Theta(\cdot, \lambda)$ that is uniquely determined by the penalty P. The general update rule for binary classification is given by:

$$\mathbf{b}^{(t+1)} = \Theta\left(\mathbf{b}^{(t)} + \eta\mathbf{X}^T\left[\mathbf{y} - \frac{1}{1+\exp(-\mathbf{X}\mathbf{b}^{(t)})}\right], \lambda\right), \tag{4.18}$$

where $\mathbf{y} \in \{0, 1\}$. The thresholding operator Θ is responsible for shrinking coefficients and, crucially, setting some of them to exactly zero, thereby performing feature selection. For our benchmark, we utilize the quantile thresholding function and the procedure effectively solves the penalized multi-class logistic regression problem by selecting K features based on the magnitude of their learned coefficients in every iteration until convergence is achieved.

4.4.5 Results

All the experiments were conducted on 11^{th} generation Intel octa-core 2.30 GHz processor.

4.4.6 Feature Selection Accuracy

Table 4.1 presents the classification accuracy of all methods on real-world datasets using CLIP features at different levels of feature sparsity. On CIFAR-10, the task is relatively simple, and most methods achieve high accuracy. However, our proposed Robust Loss Minimization method (RLM) described in Section 3.4 offers a clear efficiency advantage, reaching a peak accuracy of 91.1% with only 1500 features. This indicates a superior ability to identify a compact, highly informative feature set. The FSA and TISP baselines degrade more sharply when the number of features is heavily reduced.

On the more challenging CIFAR-100 dataset, a clear performance gap emerges. All of our proposed generative methods significantly outperform the FSA and TISP baselines, which peak at around 70.9% using 2000 features. Among the generative models, PPCA performs surprisingly

Table 4.1: Classification accuracy (%) for different methods on real datasets for CLIP features

Method	# Selected Features							
CIFAR-10, $n = 60,000, d = 2560$								
	2560	2250	2000	1750	1500	1250	1000	750
FSA	**91.1**	**91.3**	90.8	90.7	89.9	89.4	88.6	87.2
TISP	**91.1**	90.9	**91.2**	90.3	90.4	89.19	88.28	87.11
ELF	91	90.95	90.98	**90.97**	91	90.74	**90.61**	89.41
HeteroPCA	91	90.89	90.76	90.66	90.21	89.61	89.2	88.33
LFA	91	90.9	90.69	90.68	90.28	89.77	89.34	88.56
PPCA	91	90.83	90.68	90.39	90.24	89.1	88.54	87.69
Selective PCA	91	90.42	90.83	90.91	90.98	90.01	89.76	89
RLM	91	90.	90.68	90.81	**91.1**	**90.95**	90.1	**89.53**
CIFAR-100, $n = 60,000, d = 2560$								
	2560	2250	2000	1750	1500	1250	1000	750
FSA	69.63	70.5	70.09	70.92	69.43	69.28	67.72	63.24
TISP	69.63	69.94	70.82	70.42	69.36	68.58	68.29	63.25
ELF	**72.81**	72.7	72.35	72.18	71.31	70.02	68.01	65.11
HeteroPCA	**72.81**	72.01	71.51	71.36	70.28	68.9	67.06	64.51
LFA	**72.81**	72.67	72.14	72.01	71.22	70	67.09	65.01
PPCA	**72.81**	72.83	**73.01**	72.55	72.12	70.83	69.36	65.47
Selective PCA	**72.48**	72.82	72.95	72.77	**72.55**	71.02	69.21	66.23
RLM	**72.81**	**72.94**	72.95	**73.08**	72.47	**71.33**	**69.73**	**67.24**
ImageNet, $n = 1.2$ million, $d = 640$								
	640	600	550	500	450	400	350	300
FSA	71.37	71.6	71.68	71.2	69	67	65.98	64.15
TISP	71.37	71.72	71.49	70.34	69.04	67.34	65.06	63.3
ELF	**73.73**	**73.60**	73.27	**72.95**	**72.62**	**71.97**	**71.3**	**70.24**
HeteroPCA	**73.73**	73.38	73.14	72.8	72.48	71.81	71.06	69.88
LFA	**73.73**	73.49	73.23	72.94	72.53	71.94	71.11	70.14
PPCA	**73.73**	73.42	73.16	72.9	72.57	71.84	71.05	70
Selective PCA	**73.73**	73.5	73.3	72.91	72.52	71.90	71.06	70
RLM	**73.73**	73.53	**73.33**	72.94	72.55	71.87	71.19	70.22

well, achieving a peak accuracy of 73.01% using the same number of features. Our proposed RLM method also performs strongly in this group, reaching a peak of 73.08% with 1750 features only. This shows that, on complex, fine-grained tasks, our methods better preserve critical information.

On the large-scale ImageNet-1k dataset, the robustness of the generative methods is most evident. The FSA and TISP baselines suffer a severe drop in performance under aggressive pruning, falling to around 63-64% accuracy with 300 features. In contrast, all of our proposed methods remain highly stable. The ELF method is remarkably resilient, maintaining the highest accuracy of 70.24% with just 300 features. Our RLM and Selective PCA methods also deliver robust performance, confirming their suitability for large-scale, high-dimensional problems.

Table 4.2: Training time (seconds) for FSA and TISP on the datasets evaluated for CLIP features.

Dataset	Method	# Selected Features							
		2560	2250	2000	1750	1500	1250	1000	750
CIFAR-10	FSA	25	21	20	19	18	18	17	16
	TISP	25	20	20	20	19	18	17	15
	Selective PCA	52	48	48	46	45	42	40	40
	RLM	78	65	59	56	53	51	51	51
CIFAR-100	FSA	45	43	41	35	31	29	27	25
	TISP	45	44	42	37	34	28	26	25
	Selective PCA	432	420	419	415	414	410	407	401
	RLM	563	542	540	528	521	516	516	504
		640	600	550	500	450	400	350	300
ImageNet-1k	FSA	2293	1335	1329	1176	921	898	819	779
	TISP	2293	1236	1022	996	877	776	769	737
	Selective PCA	1285	1190	1124	1060	975	882	798	730
	RLM	1500	1380	1250	1130	1010	900	790	710

Table 4.3: Training time (seconds) for low-rank generative methods on the datasets evaluated for CLIP Features.

Dataset	# Features	**Methods**			
		PPCA	LFA	ELF	HeteroPCA
CIFAR-10	2560	10	30	40	58
CIFAR-100	2560	12	90	42	200
ImageNet-1k	640	46	102	248	80

Table 4.4 presents the classification accuracy results when applying our feature selection framework to the state-of-the-art DINOv3 embeddings. The use of these compelling features elevates the overall performance of all methods to a new, higher baseline, allowing us to analyze the robustness and efficiency of each selection technique in a near-optimal feature space. On the CIFAR-10 dataset, the DINOv3 features prove to be remarkably effective, pushing the accuracy for all methods close to 99%. In this high-performance scenario, the primary differentiator between methods is their ability to maintain this accuracy under aggressive feature pruning. A clear pattern emerges: the discriminative FSA and TISP methods, while achieving high peak accuracy, are the most brittle. When the feature set is reduced to just 200 (an 80% reduction), their performance collapses by over 4% points. In contrast, all generative methods demonstrated superior stability. In particular, methods such as HeteroPCA, LFA, and PPCA maintain accuracies above 98.2%, demonstrating graceful degradation. This highlights the inherent robustness of modeling class-specific distribu-

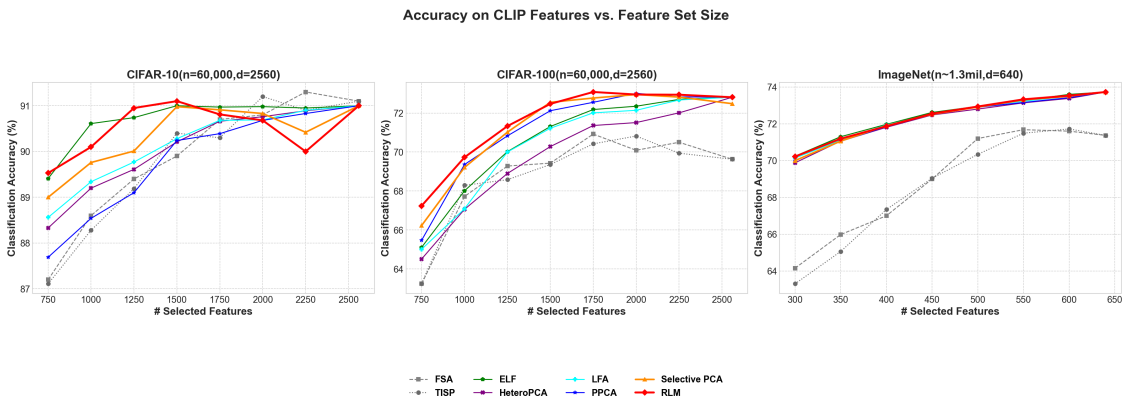


Figure 4.1: Test accuracy on real-world datasets for different methods using CLIP features

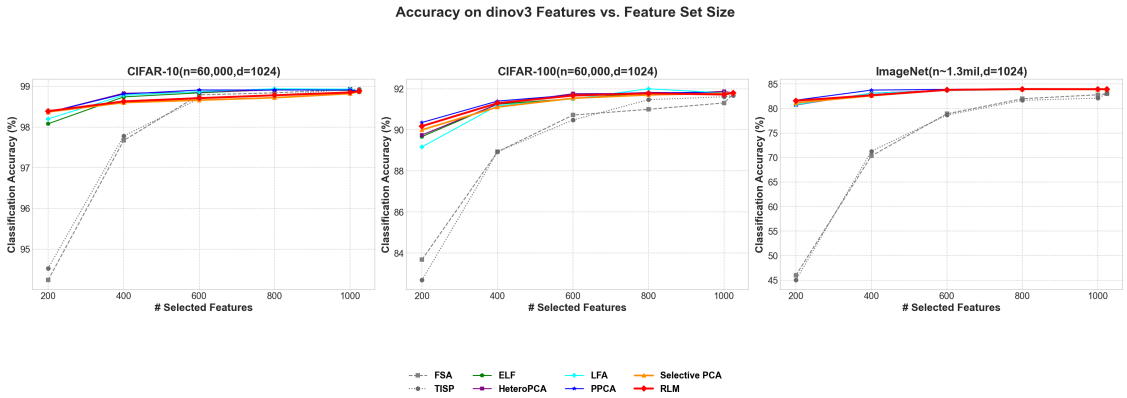


Figure 4.2: Test accuracy on real-world datasets for different methods using Dinov3 features

tions for identifying a core, information-rich feature subset. This trend is even more pronounced on the more challenging CIFAR-100 dataset. The performance gap between the discriminative baselines and the generative methods is stark. Under aggressive pruning to 200 features, FSA and TISP's accuracy plummets by over 8% points. The generative methods, however, remain remarkably stable. PPCA is the standout performer in this high-compression regime, achieving 90.36% accuracy, while our proposed RLM method is close behind at 90.18%.

This represents a performance advantage of approximately 7-8 percentage points over the discriminative baselines, providing definitive evidence that, as task complexity increases, the ability of generative models to identify stable, class-representative features is a significant advantage over methods that focus solely on the classification boundary. The LFA method achieves the highest accuracy 92% for this data using 800(78%) features out of 1024. The HeteroPCA method achieved 91.67% accuracy with only 600 (58%) features. Finally, on the large-scale ImageNet-1k bench-

Table 4.4: Classification accuracy (%) for different methods on real datasets for Dinov3 features

Method	# Selected Features					
CIFAR-10, $n = 60,000, d = 1024$						
	1024	1000	800	600	400	200
FSA	98.87	98.93	98.84	98.79	97.67	94.24
TISP	**98.93**	**98.93**	98.78	98.7	97.78	94.53
ELF	98.88	98.9	98.92	98.84	98.74	98.08
HeteroPCA	98.88	**98.93**	98.91	98.86	**98.83**	**98.36**
LFA	98.88	**98.93**	**98.94**	98.87	98.78	98.2
PPCA	98.88	98.91	98.91	**98.91**	98.81	**98.36**
Selective PCA	98.88	98.82	98.72	98.66	98.60	98.37
RLM	98.88	98.85	98.78	98.71	98.63	98.39
CIFAR-100, $n = 60,000, d = 1024$						
	1024	1000	800	600	400	200
FSA	91.74	91.31	91	90.72	88.93	83.68
TISP	91.68	91.61	91.47	90.48	88.93	82.68
ELF	**91.8**	91.8	91.8	91.52	91.25	89.67
HeteroPCA	**91.8**	**91.87**	91.77	**91.76**	91.25	89.75
LFA	**91.8**	91.8	**92**	91.51	91.18	89.17
PPCA	**91.8**	91.8	91.82	91.7	**91.4**	**90.36**
Selective PCA	**91.8**	91.78	91.70	91.54	91.11	90.0
RLM	**91.8**	91.72	91.78	91.67	91.3	90.18
ImageNet, $n = 1.2$ million, $d = 1024$						
	1024	1000	800	600	400	200
FSA	83.05	82.8	82	78.93	70.35	46
TISP	83.05	82.13	81.67	78.67	71.3	45.01
ELF	**83.93**	83.92	83.92	83.75	83.1	80.76
HeteroPCA	**83.93**	83.92	83.92	83.73	83.13	80.71
LFA	**83.93**	83.93	83.87	83.66	83.13	80.78
PPCA	**83.93**	**83.94**	**84.03**	**83.91**	**83.76**	**81.7**
Selective PCA	**83.93**	83.89	83.92	83.82	82.61	81.15
RLM	**83.93**	83.95	83.97	83.8	82.7	81.6

mark, the FSA and TISP accuracies dropped by nearly 40 percentage points when reduced to 200 (19%) features. PPCA once again demonstrates best-in-class performance, achieving its peak accuracy of 84.03% with a reduced set of 800 features—outperforming the full-feature baseline—and maintaining an impressive 81.7% accuracy with only 200 features. Our proposed RLM method

Table 4.5: Training time (seconds) for low-rank generative methods on the datasets evaluated for Dinov3 features

Dataset	# Features	**Methods**			
		PPCA	LFA	ELF	HeteroPCA
CIFAR-10	1024	5	18	9	57
CIFAR-100	1024	9	24	12	59
ImageNet	1024	133	476	916	611

Table 4.6: Training time (seconds) for FSA and TISP on the datasets evaluated for Dinov3 features.

Dataset	Method	# Selected Features					
		1024	1000	800	600	400	200
CIFAR-10	FSA	24	23	23	22	22	21
	TISP	26	26	26	25	25	24
	Selective PCA	22	21	21	20	19	18
	RLM	32	31	30	29	29	28
CIFAR-100	FSA	27	26	26	26	25	25
	TISP	30	29	28	27	26	26
	Selective PCA	175	172	168	165	162	160
	RLM	225	221	218	214	211	210
		1024	1000	800	600	400	200
ImageNet	FSA	3963	3786	2479	1222	1024	1018
	TISP	3963	3863	2326	1136	1034	1022
	Selective PCA	2060	1995	1880	1750	1640	1510
	RLM	2400	2320	2180	2020	1850	1780

closely tracks this performance, achieving 81.6% at the 200-feature level. This massive, greater-than-35-point performance advantage over the discriminative baselines confirms that for large-scale, high-dimensional problems, a feature selection strategy grounded in robust, class-specific generative modeling is overwhelmingly superior.

4.4.7 Analysis of Computational Efficiency

Tables 4.2 and 4.3 provide a comprehensive overview of the computational costs associated with the feature selection methods evaluated in this study. Table 4.2 details the training times for FSA, TISP, and our proposed Selective PCA and RLM methods. For these methods, a time is reported for each level of feature sparsity. This is because these frameworks, as evaluated here, require a separate and complete training cycle for each desired number of features. For FSA and TISP on ImageNet, a single run on the complete feature set takes 2,293 seconds (approximately 38 minutes), making a complete analysis extremely time-consuming. Our proposed methods, when

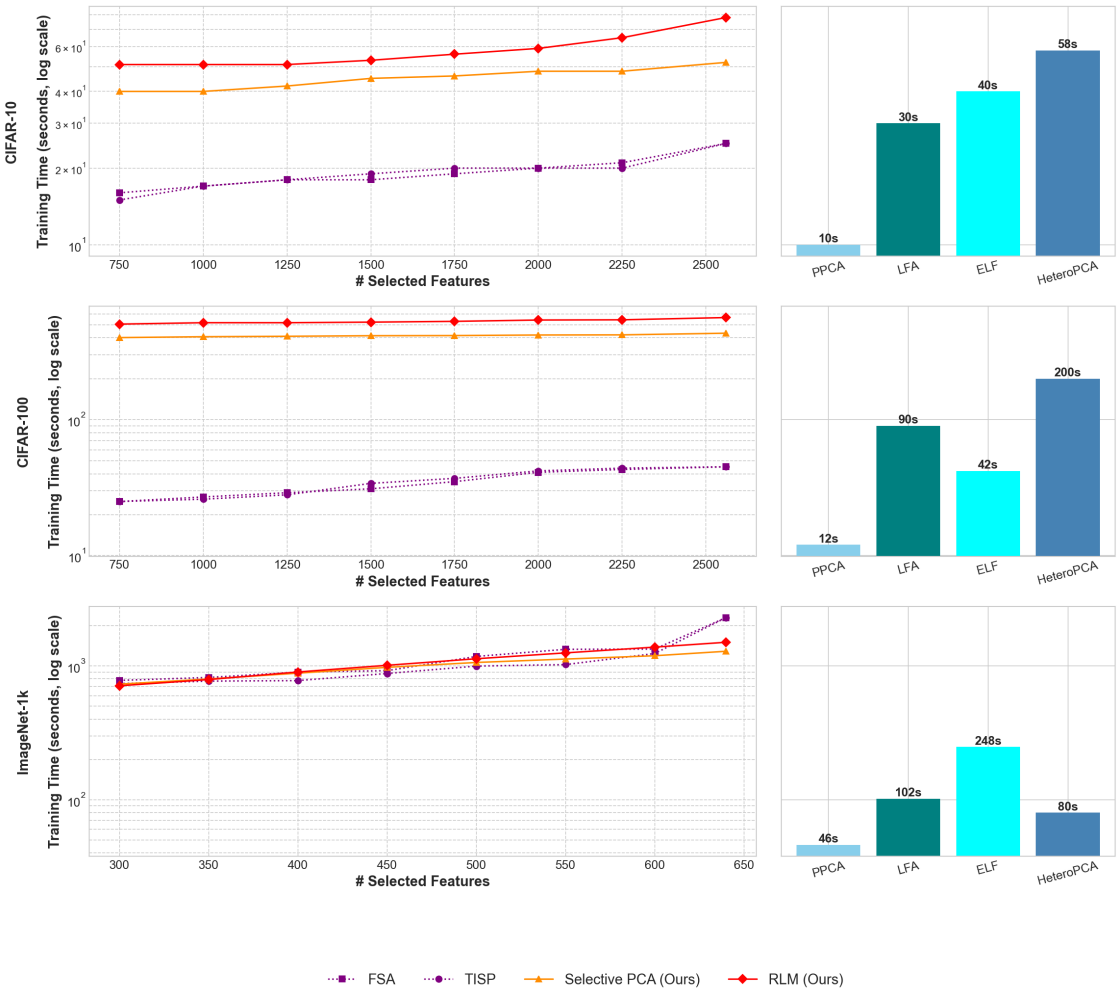


Figure 4.3: Training time (seconds) for different methods using the CLIP features

run in this same iterative manner, exhibit a similar computational profile. On ImageNet, RLM requires 1,500 seconds for its initial run, which, while faster than the full-feature FSA/TISP run, remains computationally intensive and comparable in magnitude.

In stark contrast, Table 4.3 illustrates the profound efficiency of the standard low-rank generative methods (PPCA, LFA, ELF, HeteroPCA). These methods operate on a one-shot, upfront computation paradigm. The time reported is the total time required to model the data and calculate the Signal-to-Noise Ratio (SNR) for all features once. After this single computation is complete, selecting any number of top features and updating the model parameters by restricting it to the selected features is an instantaneous operation. This architectural advantage results in a massive efficiency gain. For instance, on the ImageNet dataset, PPCA ranks all 640 features in a mere 46

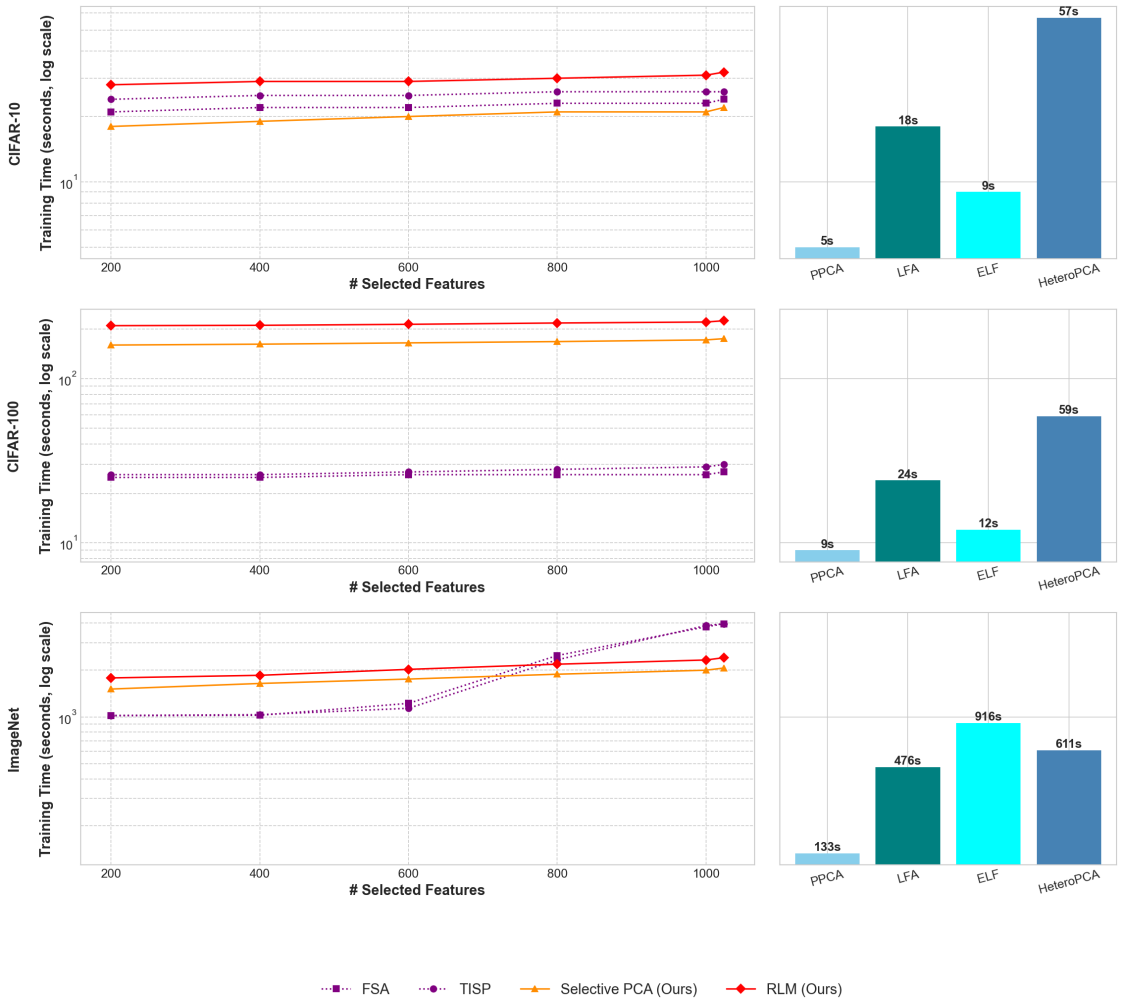


Figure 4.4: Training time (seconds) for different methods using the Dinov3 features

seconds. This is nearly 50 times faster than the 2,293 seconds required for just one of the multiple training runs needed by FSA or TISP.

Therefore, the comparison between the two tables reveals a clear trade-off. While our proposed Selective PCA and RLM methods demonstrate strong classification accuracy, their computational cost, when framed in an iterative selection process, is substantial and comparable to the expensive discriminative baselines. The standard generative methods, particularly PPCA and LFA, offer a vastly more scalable and practical solution.

The computational times for experiments using DINOv3 features are presented in Tables 4.6 and 4.5. These results confirm the same fundamental efficiency differences that were observed with the CLIP features. Table 4.5 shows that the standard generative methods (PPCA, LFA, etc.) are

very fast. They perform a single, one-time computation to rank all features. For example, PPCA completes its entire analysis for ImageNet in just 133 seconds. In contrast, Table 4.6 shows the times for the iterative methods, including our proposed Selective PCA and RLM. These methods require a separate, time-consuming training run for each feature subset. On the large ImageNet dataset, a single run of our RLM method takes 2400 seconds, and the FSA baseline takes 3963 seconds. This is substantially slower than the entire one-time analysis performed by the standard generative models, confirming that they offer a much more scalable and efficient approach for feature ranking.

4.4.8 Class Incremental Learning Experiments

Our framework is architecturally designed to be inherently immune to catastrophic forgetting. This stems from the complete decoupling of class-specific knowledge: we train an independent generative model (e.g., PPCA or LFA) for each class using only its data, and these models remain unmodified thereafter. When new classes arrive, we train separate models exclusively for them. Similarly, feature selection follows an additive paradigm—features with high SNR are identified for the new classes and appended to the overall feature pool, preserving all prior selections. Since neither the parameters of existing models nor the selected features for previous classes are altered, the framework's performance on old classes cannot degrade, ensuring zero forgetting without additional mechanisms such as replay buffers or regularization.

To validate this property, we adopt the standard CIL protocol from [47], a widely used benchmark for evaluating incremental methods [114, 128, 81]. The protocol proceeds as follows: We initialize the model on a random subset comprising half the dataset's classes (e.g., 50 for CIFAR-100, 500 for ImageNet-1K). The remaining classes are divided into 5 or 10 equal incremental tasks, introduced sequentially. At each step b:

- Train independent generative models for the new classes in the task b.
- Apply SNR-based feature selection to identify the top discriminative features for these classes.
- Add the selected set of features to the existing feature pool, and also keep track of the selected set of features for every class.

Evaluation occurs after each step on a held-out test set encompassing all classes seen thus far, using only the selected features for classification via the Mahalanobis distance. No exemplars from prior classes are stored or replayed, aligning with the exemplar-free setting [128]. To ensure robustness, we repeated the process over 5 independent runs with different random seeds for class

splitting and report the *average incremental accuracy*: the mean accuracy across all evaluation steps.

Table 4.7: Comparison of average incremental accuracy (%) and final accuracy on CIFAR-100 and ImageNet-1K in the class-incremental learning setting (exemplar-free).

Method	**CIFAR-100**				**ImageNet-1K**			
	$\bar{a}_{10}$	a_{10}	$\bar{a}_5$	a_5	$\bar{a}_{10}$	a_{10}	$\bar{a}_5$	a_5
iCaRL(2017) [89]	52.57	50	57.17	45.5	46.72	45.6	51.36	39.89
BiC(2019) [118]	53.21	-	56.86	40.21	**84.02**	73.2	-	-
UCIR (NME)(2019) [47]	60.12	-	63.12	-	59.92	-	61.56	-
UCIR (CNN)(2019) [47]	60.18	43.39	63.42	-	61.28	-	64.34	-
PODNet(2020) [30]	63.19	41.05	64.83	-	64.13	-	66.95	-
PPCA-CLIP(2023)[114]	69.71	72.81	69.71	72.81	71.25	73.73	71.25	**73.73**
LwF (2018) [70]	82.88	77.57	88.10	84.28	-	-	-	-
L2P (2022) [116]	89.48	84.47	91.02	86.27	-	-	-	-
DualPrompt (2022) [115]	88.86	84.23	89.78	84.76	-	-	-	-
ACIL (2022) [119]	91.96	90.33	**94.00**	90.73	-	-	-	-
CODA-Prompt (2023) [106]	91.19	87.24	92.20	88.67	-	-	-	-
LAE (2023) [76]	86.97	81.13	88.50	82.76	-	-	-	-
DS-AL (2024) [26]	83.50	86.05	88.82	85.91	-	-	-	-
SimpleCIL (2024) [128]	82.31	76.21	81.12	76.21	-	-	-	-
Aper (2024) [128]	90.91	85.81	91.56	87.51	-	-	-	-
EASE (2024) [71]	92.01	87.25	92.81	89.22	-	-	-	-
CLIP-(ELF+SNR)(640/640)	-	-	-	-	76	**73.73**	**76**	**73.73**
CLIP-(ELF+SNR)(1750/2560)	77.37	72.81	77.37	72.81	-	-	-	-
Dinov3-(LFA+SNR)(800/1024 features)	**93.44**	**92**	93.44	**92**	-	-	-	-

Table 4.7 provides a comprehensive and detailed comparison of our proposed frameworks against a wide array of state-of-the-art (SOTA) methods in the challenging **exemplar-free class-incremental learning setting**. The table reports two distinct performance metrics for the CIFAR-100 and the large-scale ImageNet-1K benchmarks: the **average incremental accuracy** ($\bar{a}$), which is the average performance across all incremental steps, and the **final accuracy** (a), which is the performance at the final step on all classes seen so far. These metrics are evaluated for both 5-step and 10-step incremental learning scenarios to assess performance under different learning granularities.

An analysis of historical and recent SOTA methods reveals a clear upward trend in performance over time, particularly on the **CIFAR-100** benchmark. Early methods like iCaRL and BiC established baselines in the 50-60% accuracy range. The field saw significant improvement with the advent of prompt-based learning methods designed for large PTMs, such as L2P, DualPrompt, and

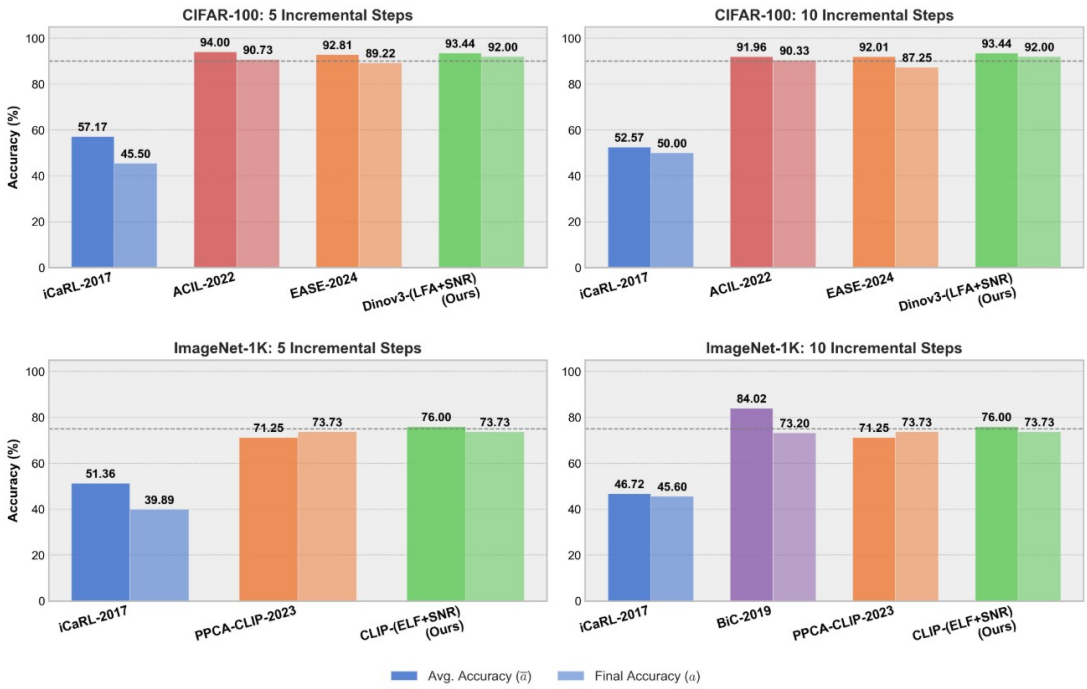


Figure 4.5: Comparison of the CIL accuracy on CIFAR100 and ImageNet-1k datasets using different methods

CODA-Prompt, which consistently pushed performance to around 90%. As of the latest results, the top-performing SOTA methods in this comparison are ACIL and EASE, with ACIL achieving the highest average accuracy ($\bar{a}_5 = 94.00\%$) and EASE achieving a strong final accuracy ($a_5 = 89.22\%$).

Against this highly competitive landscape, our 'Dinov3-(LFA+SNR)' framework demonstrates a significant performance leap, establishing a new state-of-the-art on CIFAR-100. Our method achieves a final accuracy (a_5) of **92.00%**, substantially outperforming the previous top method, EASE, by nearly 3 percentage points. Furthermore, our average accuracy ($\bar{a}_5$) of **93.44%** is highly competitive with the best reported result. Critically, this superior performance is achieved not with the full feature set, but with a **reduced subset of 800 out of 1024 features**, as determined by our SNR criterion. This simultaneously validates our feature selection methodology and demonstrates its ability to improve accuracy by removing noisy or redundant information.

On the more challenging, large-scale **ImageNet-1K** benchmark, the field is less crowded, as many recent methods do not report results for this task due to overlap between PTM training sets and ImageNet-1K. The strongest baseline in this comparison is PPCA-CLIP, which achieves an average accuracy of 71.25% and a final accuracy of 73.73%. Our 'CLIP-(ELF+SNR)' method

clearly surpasses this baseline, achieving an average accuracy ($\bar{a}_5$) of **76%**. This represents a significant improvement of nearly 5 percentage points in overall learning stability throughout the incremental process. The final accuracy (a_5) of 73.73% matches the baseline, confirming that our method maintains top-tier final performance while demonstrating substantially better learning dynamics.

In summary, the results unequivocally establish the superiority of our proposed frameworks. By synergistically combining powerful pre-trained features, robust generative modeling (LFA/ELF), and principled SNR-based feature selection, our methods advance the state of the art in exemplar-free incremental learning across both standard and large-scale benchmarks.

CHAPTER 5

CONCLUSION

This dissertation presented a comprehensive investigation into feature selection for high-dimensional data, developing a novel, theoretically-grounded, and scalable framework to address the challenges of the 'curse of dimensionality,' robustness to outliers, and class-incremental learning. Our work systematically moved from foundational principles to practical, state-of-the-art applications, delivering a unified toolkit for modern data analysis.

Our research began by establishing a feature selection methodology based on the Signal-to-Noise Ratio (SNR), derived from a family of low-rank generative models including PPCA and LFA. A cornerstone of this work was the development of rigorous theoretical guarantees, including not only asymptotic consistency results but also explicit, non-asymptotic probability bounds on the estimation errors of the signal, noise, and SNR. This theoretical analysis provides a principled foundation that moves beyond heuristic approaches, offering quantifiable confidence in the reliability of our method in practical, finite-sample scenarios.

Recognizing that real-world data is often contaminated, we then introduced a second family of methods based on sparsity-inducing penalties, culminating in a novel Robust Loss Minimization (RLM) approach. This method was specifically designed to be resilient to extreme outliers by integrating a robust loss function directly into the optimization objective.

Finally, we demonstrated the immense practical utility of our generative framework by applying it to the challenging problem of class-incremental learning (CIL). We showed that, by training independent, class-specific models, our approach is structurally immune to catastrophic forgetting, enabling seamless, scalable adaptation to new data.

Throughout this work, we explored two philosophically distinct approaches to feature selection: low-rank generative modeling and sparsity-inducing penalized models. Low-rank generative methods first learn the underlying distribution of each class's data and then use the learned parameters to perform a *post-hoc* feature ranking using the SNR. In contrast, our sparse models, such as RLM, integrate feature selection directly into the optimization objective along with a low-rank constraint, forcing the model to learn a sparse representation that minimizes a chosen loss function. While

our RLM method demonstrated unmatched robustness to outliers, a central finding of this thesis is the remarkable effectiveness of the simpler generative models.

Across extensive experiments on large-scale, real-world datasets, the **Probabilistic Principal Component Analysis (PPCA)** model consistently delivered outstanding results. It not only achieved classification accuracy that was highly competitive with — and often superior to—far more complex methods, but also did so with an incredible degree of computational efficiency. The one-shot feature ranking process of PPCA was frequently orders of magnitude faster than the iterative training required by both the discriminative baselines and our more complex robust models, making it a powerful and highly practical tool.

In the class-incremental learning setting, our generative frameworks, particularly 'Dinov3-(LFA+SNR)', established a new state of the art, validating our core thesis that a class-specific generative approach provides a superior solution for scalable, adaptive learning. Future work may focus on extending the non-asymptotic theory to our robust models and exploring hybrid methods that combine the resilience of the RLM with the profound efficiency of PPCA.

BIBLIOGRAPHY

[1] Syed Tabish Abbas and Jayanthi Sivaswamy. "Latent factor model based classification for detecting abnormalities in retinal images". In: *ACPR*. 2015, pp. 411–415. DOI: 10.1109/ACPR.2015.7486536.

[2] Yehya Abouelnaga et al. "Cifar-10: Knn-based ensemble of classifiers". In: *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2016, pp. 1192–1195.

[3] Radosław Adamczak et al. "Sharp bounds on the rate of convergence of the empirical covariance matrix". In: *Comptes Rendus. Mathématique* 349.3-4 (2011), pp. 195–200.

[4] Iftikhar Ahmad. "Feature selection using particle swarm optimization in intrusion detection". In: *International Journal of Distributed Sensor Networks* 11.10 (2015), p. 806954.

[5] Osama Ahmad Alomari et al. "Hybrid feature selection based on principal component analysis and grey wolf optimizer algorithm for Arabic news article classification". In: *IEEE Access* 10 (2022), pp. 121816–121830.

[6] Theodore Wilbur Anderson. "Asymptotic theory for principal component analysis". In: *Ann. Math. Stat.* 34.1 (1963), pp. 122–148.

[7] Andreas Antoniades and Clive Cheong Took. "Speeding up feature selection: A deep-inspired network pruning algorithm". In: *IJCNN*. 2016, pp. 360–366. DOI: 10.1109/IJCNN.2016.7727221.

[8] Rashad Aziz. "Sparse Methods for Latent Class Analysis, Principal Component Analysis and Regression with Missing Data". PhD thesis. The Florida State University, 2023.

[9] Jushan Bai and Kunpeng Li. "STATISTICAL ANALYSIS OF FACTOR MODELS OF HIGH DIMENSION". In: *Ann. Stat.* (2012), pp. 436–465.

[10] Zhidong Bai and Jian-feng Yao. "Central limit theorems for eigenvalues in a spiked population model". In: *Annales de l'IHP Probabilités et statistiques*. Vol. 44. 2008, pp. 447–474.

[11] Jinho Baik and Jack W Silverstein. "Eigenvalues of large sample covariance matrices of spiked population models". In: *Journal of multivariate analysis* 97.6 (2006), pp. 1382–1408.

[12] N Nasrin Banu and Radha Senthil Kumar. "Online Feature Selection Using Sparse Gradient". In: *International Journal on Artificial Intelligence Tools* 31.08 (2022), p. 2250038.

[13] Adrian Barbu et al. "Feature selection with annealing for computer vision and big data learning". In: *IEEE Trans. on PAMI* 39.2 (2017), pp. 272–286.

[14] Jonathan T. Barron. *A General and Adaptive Robust Loss Function*. 2019. arXiv: 1701.03077 [cs.CV].

[15] Kenneth W Bauer Jr, Stephen G Alsing, and Kelly A Greene. "Feature screening using signal-to-noise ratios". In: *Neurocomputing* 31.1-4 (2000), pp. 29–44.

[16] Trevor J Bihl, William A Young II, and Adam Moyer. "Physics-Informed Feature Engineering and R2-Based Signal-to-Noise Ratio Feature Selection to Predict Concrete Shear Strength". In: *Mathematics* (2025).

[17] Michael J Black and Paul Anandan. "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields". In: *CVIU* 63.1 (1996), pp. 75–104.

[18] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. "Unsupervised feature selection for principal components analysis". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 61–69.

[19] Florentina Bunea, Yiyuan She, and Marten H Wegkamp. "Joint variable and rank selection for parsimonious estimation of high-dimensional matrices". In: (2012).

[20] Florentina Bunea and Luo Xiao. "On the sample covariance matrix estimator of reduced effective rank population matrices, with applications to fPCA". In: (2015).

[21] T Tony Cai, Zongming Ma, and Yihong Wu. "Sparse PCA: Optimal rates and adaptive estimation". In: (2013).

[22] T Tony Cai, Cun-Hui Zhang, and Harrison H Zhou. "Optimal rates of convergence for covariance matrix estimation". In: (2010).

[23] Ankur Chanda. "An In-Depth Analysis of CIFAR-100 Using Inception v3". In: ().

[24] Girish Chandrashekar and Ferat Sahin. "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[25] Pierre Charbonnier et al. "Two deterministic half-quadratic regularization algorithms for computed imaging". In: *ICIP*. Vol. 2. IEEE. 1994, pp. 168–172.

[26] Y. Chen et al. "Dynamic Subspace Adaptation for Continual Learning". In: *AAAI* (2024).

[27] John E Dennis and Roy E Welsch. "Techniques for nonlinear least squares and robust regression". In: *Communications in Statistics - Simulation and Computation* 7.4 (1978), pp. 345–359.

[28] Prithviraj Dhar et al. "Learning without memorizing". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5138–5146.

[29] Alexey Dosovitskiy. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[30] Arthur Douillard et al. "Podnet: Pooled outputs distillation for small-tasks incremental learning". In: *European Conference on Computer Vision*. Springer. 2020, pp. 86–102.

[31] Jianqing Fan and Runze Li. "Variable selection via nonconcave penalized likelihood and its oracle properties". In: *Journal of the American statistical Association* 96.456 (2001), pp. 1348–1360.

[32] Laura Florescu and Will Perkins. "Spectral thresholds in the bipartite stochastic block model". In: *COLT*. PMLR. 2016, pp. 943–959.

[33] Zijian Gao et al. "Knowledge Memorization and Rumination for Pre-trained Model-based Class-Incremental Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2025, pp. 20523–20533.

[34] Donald Geman and Stuart Geman. "Bayesian image analysis". In: *Disordered systems and biological organization*. Springer, 1986, pp. 301–319.

[35] Zoubin Ghahramani, Geoffrey E Hinton, et al. *The EM algorithm for mixtures of factor analyzers*. Tech. rep. CRG-TR-96-1, University of Toronto, 1996.

[36] MA Girshick. "On the sampling theory of roots of determinantal equations". In: *Ann. Math. Stat.* 10.3 (1939), pp. 203–224.

[37] MA Girshick. "Principal components". In: *JASA* 31.195 (1936), pp. 519–528.

[38] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[39] Yangzi Guo, Yiyuan She, and Adrian Barbu. "Network pruning via annealing and direct sparsity control". In: *IJCNN*. 2021, pp. 1–8.

[40] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *JMLR* 3.3 (2003), pp. 1157–1182.

[41] Moussa Hamadache and Dongik Lee. "Principal component analysis based signal-to-noise ratio improvement for inchoate faulty signals: Application to ball bearing fault detection". In: *International Journal of Control, Automation and Systems* 15.2 (2017), pp. 506–517.

[42] Muhammad Hamraz et al. "Feature selection for high dimensional microarray gene expression data via weighted signal to noise ratio". In: *PloS one* 18.4 (2023), e0284619.

[43] Ke Han and Adrian Barbu. "Semi-Supervised Few-Shot Incremental Learning with k-Probabilistic Principal Component Analysis". In: *Electronics* 13.24 (2024). ISSN: 2079-9292. DOI: 10.3390/electronics13245000. URL: https://www.mdpi.com/2079-9292/13/24/5000.

[44] Douglas M Hawkins and David Olive. "Applications and algorithms for least trimmed sum of absolute deviations regression". In: *Computational Statistics & Data Analysis* 32.2 (1999), pp. 119–134.

[45] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[46] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[47] Saihui Hou et al. "Learning a unified classifier incrementally via rebalancing". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 831–839.

[48] Chenn-Jung Huang and Wei-Chen Liao. "A comparative study of feature selection methods for probabilistic neural networks in cancer classification". In: *Int. Conf. on Tools with AI*. IEEE. 2003, pp. 451–458.

[49] Hai-Hui Huang, Xiao-Ying Liu, and Yong Liang. "Feature selection and cancer classification via sparse logistic regression with the hybrid L1/2+ 2 regularization". In: *PloS one* 11.5 (2016), e0149675.

[50] Peter J Huber. "A robust version of the probability ratio test". In: *Ann. Math. Stat.* (1965), pp. 1753–1758.

[51] Peter J Huber. "Finite Sample Breakdown of M-and P-Estimators". In: *Ann. Stat.* 12.1 (1984), pp. 119–126.

[52] Peter J Huber. "Robust estimation of a location parameter". In: *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 492–518.

[53] Ibrar Hussain et al. "Optimal features selection in the high dimensional data based on robust technique: Application to different health database". In: *Heliyon* 10.17 (2024).

[54] Alan J Izenman. *Modern multivariate statistical techniques*. Vol. 1. Springer, 2008.

[55] MOHD IZHAR et al. "Adapting VGG16: Exploring Strategies for Real-World Image Classification Challenges using CIFAR-10 to CIFAR-100". In: (2025).

[56] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: A review". In: *IEEE Trans. on PAMI* 22.1 (2000), pp. 4–37.

[57] I. M. Johnstone and A. Y. Lu. "On consistency and sparsity for principal components analysis in high dimensions". In: *Journal of the American Statistical Association* 104.486 (2009), pp. 682–693.

[58] Iain M Johnstone. "On the distribution of the largest eigenvalue in principal components analysis". In: *Ann. Stat.* 29.2 (2001), pp. 295–327.

[59] Rittwika Kansabanik and Adrian Barbu. "Feature Selection for Latent Factor Models". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 30742–30751.

[60] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. "A survey of feature selection and feature extraction techniques in machine learning". In: *Science and information conference*. IEEE. 2014, pp. 372–378.

[61] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[62] V. Koltchinskii and K. Lounici. "Concentration inequalities and moment bounds for sample covariance operators". In: *Bernoulli* 23.1 (2017), pp. 110–133.

[63] Vladimir Koltchinskii and Karim Lounici. "Normal approximation and concentration of spectral projectors of sample covariance". In: (2017).

[64] Alex Krizhevsky, Geoff Hinton, et al. "Convolutional deep belief networks on cifar-10". In: *Unpublished manuscript* 40.7 (2010), pp. 1–9.

[65] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR-10 (Canadian Institute for Advanced Research)*. Tech. rep. University of Toronto, 2009. URL: `https://www.cs.toronto.edu/~kriz/cifar.html`.

[66] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR-100 (Canadian Institute for Advanced Research)*. Tech. rep. University of Toronto, 2009. URL: `https://www.cs.toronto.edu/~kriz/cifar.html`.

[67] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[68] Yvan G Leclerc. "Constructing simple stable descriptions for image partitioning". In: *IJCV* 3.1 (1989), pp. 73–102.

[69] Seunggeun Lee, Fei Zou, and Fred A Wright. "Convergence and prediction of principal component scores in high-dimensional settings". In: *Ann. Stat.* 38.6 (2010), p. 3605.

[70] Z. Li and D. Hoiem. "Learning without Forgetting". In: *IEEE TPAMI* (2018).

[71] Z. Li et al. "EASE: Efficient and Stable Continual Learning with Adapters". In: *CVPR* (2024).

[72] Hyunki Lim. "Low-rank learning for feature selection in multi-label classification". In: *Pattern Recognition Letters* 172 (2023), pp. 106–112.

[73] Che Liu et al. "Does DINOv3 Set a New Medical Vision Standard?" In: *arXiv preprint arXiv:2509.06467* (2025).

[74] Huan Liu and Hiroshi Motoda. *Computational methods of feature selection*. CRC press, 2007.

[75] Jiarun Liu et al. "Swin-umamba: Mamba-based unet with imagenet-based pretraining". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2024, pp. 615–625.

[76] Y. Liu et al. "Learning from Ambiguous Examples for Incremental Learning". In: *ICCV* (2023).

[77] Karim Lounici et al. "Oracle inequalities and optimal inference under group sparsity". In: (2011).

[78] Jasin Machkour et al. *Sparse PCA with False Discovery Rate Controlled Variable Selection*. 2024. arXiv: `2401.08375 [stat.ML]`. URL: `https://arxiv.org/abs/2401.08375`.

[79] Debahuti Mishra and Barnali Sahu. "Feature selection for cancer classification: a signal-to-noise ratio approach". In: *Int. J. of Scientific & Eng. Research* 2.4 (2011), pp. 1–7.

[80] Boaz Nadler. "Finite Sample Approximation Results for Principal Component Analysis: A Matrix Perturbation Approach". In: *Ann. Stat.* (2008), pp. 2791–2817.

[81] Kengo Nakata et al. "Revisiting a knn-based image classification system with high-capacity storage". In: *European conference on computer vision*. Springer. 2022, pp. 457–474.

[82] Zhiguo Niu and Xuehong Qiu. "Facial expression recognition based on weighted principal component analysis and support vector machines". In: *Int. Conf. on Advanced Computer Theory and Engineering*. Vol. 3. IEEE. 2010, pp. V3–174.

[83] Myeung Suk Oh et al. "Minimum description feature selection for complexity reduction in machine learning-based wireless positioning". In: *IEEE Journal on Selected Areas in Communications* 42.9 (2024), pp. 2585–2600.

[84] Debashis Paul. "Asymptotics of sample eigenstructure for a large dimensional spiked covariance model". In: *Statistica Sinica* (2007), pp. 1617–1642.

[85] L Pushpalatha and R Durga. "Cardio Vascular Disease Prediction Based on PCA-ReliefF Hybrid Feature Selection Method with SVM". In: *International Conference on Advancements in Smart Computing and Information Security*. Springer. 2023, pp. 40–54.

[86] Alec Radford et al. *CLIP: Connecting Text and Images*. `https://github.com/openai/CLIP`. Accessed: ¡2024-05-05¿. 2021.

[87] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *ICML*. PMLR. 2021, pp. 8748–8763.

[88] Fariq Rahmat et al. "Supervised feature selection using principal component analysis". In: *Knowledge and Information Systems* 66.3 (2024), pp. 1955–1995.

[89] Sylvestre-Alvise Rebuffi et al. "icarl: Incremental classifier and representation learning". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010.

[90] Benjamin Recht et al. "Do cifar-10 classifiers generalize to cifar-10?" In: *arXiv preprint arXiv:1806.00451* (2018).

[91] Peter J Rousseeuw and Annick M Leroy. "A robust scale estimator based on the shortest half". In: *Statistica Neerlandica* 42.2 (1988), pp. 103–116.

[92] Donald B Rubin and Dorothy T Thayer. "EM algorithms for ML factor analysis". In: *Psychometrika* 47 (1982), pp. 69–76.

[93] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[94] Olga Russakovsky et al. "Img.Net Large Scale Visual Recognition Challenge". In: *IJCV* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[95] Barnali Sahu, Satchidananda Dehuri, and Alok Kumar Jagadev. "Feature selection model based on clustering and ranking in pipeline for microarray data". In: *Informatics in Medicine Unlocked* 9 (2017), pp. 107–122.

[96] Diego Salas-Gonzalez et al. "Feature selection using factor analysis for Alzheimer's diagnosis using PET images". In: *Medical physics* 37.11 (2010), pp. 6084–6095.

[97] Matthew Shardlow. "An analysis of feature selection techniques". In: *The University of Manchester* 1.2016 (2016), pp. 1–7.

[98] Ali Sharif Razavian et al. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 806–813.

[99] Sara Sharifzadeh et al. "Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection". In: *Engineering Applications of Artificial Intelligence* 65 (2017), pp. 168–177.

[100] Neha Sharma, Vibhor Jain, and Anju Mishra. "An analysis of convolutional neural networks for image classification". In: *Procedia computer science* 132 (2018), pp. 377–384.

[101] Yiyuan She. "An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors". In: *Computational Statistics & Data Analysis* 56.10 (2012), pp. 2976–2990.

[102] Yiyuan She. "Selective factor extraction in high dimensions". In: *Biometrika* 104.1 (2017), pp. 97–110.

[103] Yiyuan She. "Thresholding-based iterative selection procedures for model selection and shrinkage". In: *Elec. J. of Stat.* 3 (2009), pp. 384–415.

[104] Tahira Shehzadi et al. "Object detection with transformers: A review". In: *Sensors* 25.19 (2025), p. 6025.

[105] Oriane Siméoni et al. "Dinov3". In: *arXiv preprint arXiv:2508.10104* (2025).

[106] J. S. Smith et al. "CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning". In: *CVPR* (2023).

[107] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5227–5237.

[108] Jiliang Tang, Salem Alelyani, and Huan Liu. "Feature selection for classification: A review". In: *Data classification: Algorithms and applications* (2014), p. 37.

[109] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *JRSS B* 58.1 (1996), pp. 267–288.

[110] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". In: *JRSS B* 61.3 (1999), pp. 611–622.

[111] F William Townes et al. "Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model". In: *Genome biology* 20.1 (2019), p. 295.

[112] Harun Uğuz. "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm". In: *Knowledge-Based Systems* 24.7 (2011), pp. 1024–1032.

[113] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press, 2019.

[114] Boshi Wang and Adrian Barbu. "Scalable Learning with Incremental Probabilistic PCA". In: *IEEE Int. Conf. on Big Data*. 2022, pp. 5615–5622. DOI: 10.1109/BigData55660.2022.10020330.

[115] Z. Wang et al. "DualPrompt: Complementary Prompting for Rehearsal-free Continual Learning". In: *ECCV* (2022).

[116] Z. Wang et al. "Learning to Prompt for Continual Learning". In: *CVPR* (2022).

[117] Christine M Waternaux. "Asymptotic distribution of the sample roots for a nonnormal population". In: *Biometrika* 63.3 (1976), pp. 639–645.

[118] Yue Wu et al. “Large scale incremental learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 374–382.

[119] H.-J. Ye et al. “Adversarial Consistency for Incremental Learning”. In: *NeurIPS* (2022).

[120] Shipeng Yu et al. “Supervised probabilistic principal component analysis”. In: *SIGKDD*. 2006, pp. 464–473.

[121] Sergey Zagoruyko and Nikos Komodakis. “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer”. In: *arXiv preprint arXiv:1612.03928* (2016).

[122] Anru R Zhang, T Tony Cai, and Yihong Wu. “Heteroskedastic PCA: Algorithm, optimality, and applications”. In: *Ann. Stat.* 50.1 (2022), pp. 53–80.

[123] Zhanquan Zhang et al. “Effects of 1-methylcyclopropene (1-MCP) on ripening and resistance of jujube (Zizyphus jujuba cv. Huping) fruit against postharvest disease”. In: *LWT-Food science and technology* 45.1 (2012), pp. 13–19.

[124] Junjing Zheng et al. *Fast Sparse PCA via Positive Semidefinite Projection for Unsupervised Feature Selection*. 2023. arXiv: 2309.06202 [cs.CV]. URL: https://arxiv.org/abs/2309.06202.

[125] YuYu Zheng, HaoXuan Huang, and JunMing Chen. “Comparative analysis of various models for image classification on Cifar-100 dataset”. In: *Journal of Physics: Conference Series*. Vol. 2711. 1. IOP Publishing. 2024, p. 012015.

[126] Nikita Zhivotovskiy. “Dimension-free bounds for sums of independent matrices and simple tensors via the variational principle”. In: *Electronic Journal of Probability* 29 (2024), pp. 1–28.

[127] Changjun Zhou et al. “Face recognition based on PCA and logistic regression analysis”. In: *Optik* 125.20 (2014), pp. 5916–5919.

[128] Da-Wei Zhou et al. “Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need”. In: *International Journal of Computer Vision* 133.3 (2025), pp. 1012–1032.

[129] Xiangyang Zhu et al. “Not all features matter: Enhancing few-shot clip with adaptive prior refinement”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 2605–2615.

[130] Hui Zou, Trevor Hastie, and Robert Tibshirani. “Sparse principal component analysis”. In: *JCGS* 15.2 (2006), pp. 265–286.

BIOGRAPHICAL SKETCH

Rittwika Kansabanik was born in Kolkata, West Bengal, India. Rittwika Kansabanik began her journey in Statistics at Presidency University in Kolkata, where she earned a Bachelor of Science with First Class Honors in Mathematics and Statistics. Her undergraduate work culminated in a capstone project analyzing stock market volatility using GARCH models. She continued her advanced training at the prestigious Indian Statistical Institute in Kolkata, earning her Master of Science in Statistics in 2018. During her master's studies, her thesis focused on Bayesian modeling, where she analyzed the Challenger Space Shuttle data and investigated gender disparities in household resource allocation.

In August 2020, Rittwika joined the Department of Statistics at Florida State University to pursue her Doctor of Philosophy under the supervision of Professor Adrian Barbu. Her doctoral research has focused on the critical intersection of high-dimensional statistics and machine learning, with a primary emphasis on developing novel, theoretically grounded methods for feature selection.

Her dissertation, "SNR-based Feature Selection using Low-rank Generative Models for High-dimensional Data," introduces an innovative framework that leverages the Signal-to-Noise Ratio (SNR) within class-specific generative models. A key contribution of this work is the establishment of rigorous theoretical guarantees for true feature recovery, providing a principled foundation that moves beyond heuristic approaches. Her broader research interests have also led her to develop new methods for robust variable screening in the presence of heavy-tailed outliers and to engineer Agentic AI platforms for biomedical research using Large Language Models (LLMs).

Rittwika's contributions to the field have been recognized through both publications and presentations at premier academic venues. Her research has culminated in a first-author publication at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), a top-tier highly selective conference. She has also disseminated her work widely, presenting at leading scientific meetings including the Joint Statistical Meetings (JSM), the Southern Regional Council on Statistics (SRCOS), and the Quality and Productivity Research Conference (QPRC).

Her academic excellence and research promise have been acknowledged with several honors. At Florida State University, she received the Best First-Year Ph.D. Student Award and secured the 2nd Rank in the Ph.D. Qualifying Exam. She is also a recipient of the prestigious INSPIRE Scholarship from the Department of Science and Technology (Government of India), which selects

only the top 1% of students each year. Complementing her research, she gained valuable industry experience as a Data Science Associate at PwC, where her performance was recognized with the "We Applaud" award. Passionate about applying her statistical expertise to high-impact problems, upon graduation, Rittwika will join the Stanford Cancer Research Center as a Postdoctoral associate.